Project no.:	**027657**
Project full title:	**Perception, Action & Cognition through learning of Object-Action Complexes**
Project Acronym:	**PACO-PLUS**
Deliverable no.:	**D5.2.3**
Title of the deliverable:	**Technical Report: Grammar Induction & Parsing**

Contractual Date of Delivery to the CEC:	31 January 2010	
Actual Date of Delivery to the CEC:	17 June 2010	
Organisation name of lead contractor for this deliverable:	UEDIN	
Author(s): Mark Steedman		
Participant(s): UEDIN		
Work package contributing to the deliverable:	WP5	
Nature:	R	
Version:	Final	
Total number of pages:	40	
Start date of project:	1st Feb. 2006	**Duration:** 52 month

Project co-funded by the European Commission within the Sixth Framework Programme (2002–2006)
Dissemination Level

PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract:

The core focus of workpackage WP5.2 is to link the non-linguistic Object-Action Complex (OAC)-based conceptual representation developed under the PACO-PLUS project to language via a universal Language Acquisition Algorithm, and to deploy the learned grammar in the task of understanding and generating purposeful dialog. As with human children, the conceptual representation that our systems induce from interaction with the world via low-level continuous control systems, such as the SDU robot/vision system in WP4.1, are language-independent. The language acquisition algorithm must therefore be capable of learning the syntax of any human language from exposure to utterances pairing such conceptual representations (among noise and distractors) with the appropriate sentence in that language, with the conceptual representation providing the semantics. Different languages partition that conceptual content into syntactic units such as word- and phrase-meaning pairs in different ways. So our learning algorithm must consider all such partitions. Thus, the basic idea can be summed up as saying that the child acquires language by learning a parsing model for universal grammar, of the same kind used in the wide-coverage parsers developed in prior work. Some of the work under WP5.2 concerns the evaluation of those parsers in comparison to other publicly available parsers and as models of human sentence processing.

Keyword list: Grammar Induction, Grounded Language Acquisition, Parsing

Table of Contents

1. Executive Summary

The core focus of workpackage WP5.2 is to link the non-linguistic Object-Action Complex (OAC)-based conceptual representation developed under the PACO-PLUS project to language via a universal Language Acquisition Algorithm, based on the wide coverage CCG parsers developed under other funding. Dialog applications are reported elsewhere in Workpackage 5.1. s0791051

1.1 The Place of Language in the PACO-PLUS Project

As the proposal and Annex make clear, the role of language in the PACO-PLUS project is not primarily to act as a real-time user interface to the various robot platforms involved. Since the repertory of high-level actions, plans, and goals of the platforms will remain quite restricted, commercial speech recognition treating the identification of the user's utterances as a finite classification problem, encoding those states and actions. is always going to be adequate, and much faster and more reliable than full blown syntactic analysis and semantic interpretation, especially in the face of the high word error rates that can be expected from state-of-the-art speech recognition used as an input for parsing.

The place of language in the PACO-PLUS project is, rather, a theoretical investigation into the nature of language itself, and its ontogeny in human child-language acquirers in prelinguistic sensory-motor cognition, planning, and the Object-Action Complex (OAC) based knowledge representation developed elsewhere in the project. While we apply this theory to an artificially constructed corpus of utterances, a substantial emphasis on human language acquisition is involved.

The current deliverable, as specified in the Detailed Implementation Plan, is principally concerned with real data of child-directed speech, and its relation to the differently-grounded artificial corpus. A substantial amount of work was reported in the period up to M36 on transforming the dependency-annotated part of the CHILDES corpus into a quasi-semantic representation for this purpose. Since that time we have under Task 5.2.1 developed a general purpose language learner for that corpus. We have also applied the learner to a newly available version of the GeoQuery database of question/database-query pairs (Wong and Mooney, 2007) that includes multilingual versions of the questions. We have also under Task 5.2.2 evaluated the wide coverage CCG parser developed in prior work in comparison to the State of the Art, and as a model of human sentence processing.

1.2 Task 5.2.1 Grammar Induction

The aim of this task is to build a general model of child language acquisition, taking strings in any language paired with universal logical forms as input, and yielding an incrementally growing language-specific lexicon and parsing model.

We have completed the adaptation of the CHILDES corpus "Eve" annotation of 34 hours of English child-directed utterances with dependency-structures (Sagae et al., 2007) to act as pseudo-logical forms for grammar induction. Such structures are to some extent language-specific, but by ignoring linear ordering information we can simulate a slightly simplified version of the childs problem of deciding which substructures of the logical form are lexicalized, and what word-orders they encode as directional information.

Preliminary results show a significant grammar learning effect over the baseline of simple rote learning of all previously encountered string-meaning pairs, using a generative parsing model (Kwiatkowksi, Goldwater and Steedman 2009). 34 hrs of conversation is far less data than real children are exposed to, and there are problems in adapting the CHILDES annotation to this purpose, so overall performance is still quite low.

Nevertheless, we have taken the first step in building a universal model of first language acquisition using realistic data.

We have also applied similar techniques to the GeoQuery corpus of 880 English questions concerning a geographical database and the corresponding query language terms (Wong and Mooney 2007). This artificial corpus has been used for grammar induction by a number of groups, including Mooney's own, and Zettlemoyer and Collins (2005). However, their algorithms do not attempt the generality that we are trying to achieve. Zettlemoyer has been visiting Edinburgh in 2009/2010 under separate NSF funding to collaborate on this problem.

The sentences and logical forms in GeoQuery are long, and the combinatorics of the algorithm that works for CHILDES scale data make it a much harder task. Techniques to work at this scale are still under development. However, a major attraction of the GeoQuery corpus is that a 250 word subset has been been translated into Turkish, Spanish, and Japanese (Lu et al. 2008). We have also translated the entire corpus ourselves into Greek, German, and Thai. We are in the process of running the new language learner on these datasets and a paper has been submitted (Kwiatkowksi et al. 2010).

All of this work is in principle applicable to the PACO demonstrator domains and platforms. We have made substantial progress in inducing distinctively grounded action concepts from robot explorations and observation of change for the Odense robot (Mourão, Petrick and Steedman 2009) and have proved the scalability of the system using artificial planning domains taken from the annual AI planning competition (Mourão, Petrick and Steedman 2010). This work is reported separately under D5.1.3, together with the results of applying the PKS planner to robot dialogue management.

In theory, this knowledge representation could provide a suitable basis for a linguistic semantics of objects and actions as an input to the existing language learners. However, the robot itself has a minute repertoire of actions (varieties of grasp), plans (clearing up) and objects (cups), and the problems of collecting enough data from it to work with, even in a non-interactive mode to acquire the action representation, are debilitating. We have not attempted to connect the existing language learners (as distinct from the conversational agent) with existing grounded robot action representations. We will continue to work on this very hard problem, whose nature is discussed at length in Steedman 2010a,b, and in Appendix A to the present proposal. However, for the purposes of the PACO project's stated aims, we regard the goal of proving multilingual capability for realistic datasets as a higher priority.

1.3 Task 5.2.2 Parsing

The above induced grammars are expressed as required in a format compatible with the UEDIN OpenCCG parser/generator. However, the main problem with the learner is scaling to larger grammars and the long sentences of the Geoquery corpus, which requires using the model in beam-search or Gibbs-sampling modes that OpenCCG is not well adapted to or efficient enough for. The learner therefore remains transparent to OpenCCG, rather than being expressed in OpenCCG as such.

Related parsers have been developed in OpenCCG to explore the separate problem of generalizing the Hockenmaier wide-coverage parser and parsing model using unlabeled data, as proposed in the Technical Annex. An empirical investigation of the relation between these parsers and human psycholinguistic performance is published as Morgan, Keller and Steedman 2010. A comparison between CCG wide coverage parsers and other available parsers appears as Rimell, Clark and Steedman 2009. This work will continue under separate ERC Advanced Fellowship funding obtained by Steedman.

1.4 Conclusion

A number of significant developments are reported in the publications below:

- A significant grammar learning effect over a baseline of simply recalling all previously encountered string-meaning pairs, is shown, using a generative parsing model over the English CHILDES data (Kwiatkowski, Goldwater and Steedman, 2009—see B below)
- A significant multilingual grammar learning effect over the baseline is shown for a number of languages other than English (Kwiatkowski *et al.* 2010—see C below).
- CCG parsing is shown to be State of the Art in wide coverage parsing, with specific strengths in long-range dependencies relevant to question answering—see D below.
- Head dependency models routinely used in wide coverage parsing are shown to predict human language processing difficulty—see E below

A number of questions remain open at the time of this report and constitute further work.

- How can the low-level machine-learned robot action representations of D5.1.3 be linked to a high-level natural language semantics to support non-trivial grounded language learning of the kind reported here?
- How can the interpersonal semantics of mutual knowledge be brought into this learning?
- How can the complexity of building parsing models for universal grammar be better contained? By better statistical methods such as sampling? Or by stronger constraints on the search space?

2. References

2.1 New Publications Associated with D5.2.3

Kwiatkowksi, Tom, Sharon Goldwater, and Mark Steedman. 2009. "Computational Grammar Acquisition from CHILDES data using a Probabilistic Parsing Model." In *Workshop on Psycho-Computational Models of Human Language Acquisition, at the 31st Annual Meeting of the Cognitive Science Society*.

Kwiatkowksi, Tom, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. "Probabilistic CCG Grammar Induction using Higher-Order Unification." In *Conference on Empirical Methods in Natural Language Processing*. (to be submitted).

Morgan, Emily, Frank Keller, and Mark Steedman. 2010. "A Bottom-Up Parsing Model of Local Coherence Effects." In *Proceedings of the 32nd Annual Meeting of the Cognitive Science Society*. Portland, OR: Cognitive Science Society.

Rimell, Laura, Stephen Clark, and Mark Steedman. 2009. "Unbounded Dependency Recovery for Parser Evaluation." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 813–821. Singapore: Association for Computational Linguistics.

Steedman, Mark. 2010a. *The Natural Semantics of Scope*. Cambridge, MA: MIT Press. (accepted for publication).

Steedman, Mark. 2010b. "Romantics and Revolutionaries." *Linguistics in Language Technology*, 4. (to appear).

2.2 Other References

Lu, Wei, Hwee Tou Ng, Wee Sun Lee, and Luke S. Zettlemoyer. 2008. "A Generative Model for Parsing Natural Language to Meaning Representations." In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 783–792. Association for Computational Linguistics.

Mourão, Kira, Ronald Petrick, and Mark Steedman. 2009. "Learning Action Effects in Partially Observable Domains." In *Proceedings of the ICAPS 2009 Workshop on Planning and Learning*, 15–22. Thessaloniki, Greece.

Mourão, Kira, Ronald Petrick, and Mark Steedman. 2010. "Learning Action Effects in Partially Observable Domains." In *Proceedings of the 19th European Conference on AI*. Lisbon.

Wong, Yuk Wah, and Raymond Mooney. 2007. "Learning Synchronous Grammars for Semantic Parsing with Lambda Calculus." In *Proceedings of t and he 45th Annual Meeting of the Association for Computational Linguistics*, 960–967. ACL.

Zettlemoyer, Luke, and Michael Collins. 2005. "Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars." In *Proceedings of the 21st Conference on Uncertainty in AI (UAI)*, 658–666. ACL.

3. APPENDICES

A number of additional documents are also attached to this deliverable. Here we briefly sketch the relation of each paper to this workpackage and deliverable, and make links to the specific contribution of each paper.

[A] *Report*: The report provides a new definition of the relation of the action representation to the theory of grammar, unifying the two systems theoretically via a shared characteristic automaton, the Embedded Push-Down Automaton (EPDA).

[B] *Kwiatkowski et al. (2009)*: The paper reports the application of the statistical language learning model to the transformed CHILDES data.

[C] *Kwiatkowski et al. (2010)*: The paper reports the application of the statistical language learning model to the English, Turkish, Spanish and Japanese version of the GeoQuery database.

[D] *Rimell et al. (2009)*: The paper presents a detailed comparison and error analysis of five publicly available parsers, including our own, applied "out of the box" on seven challenging diagnostic constructions.

[E] *Morgan et al. (2010)*: The paper uses a probabilistic head dependency parsing model from our wide coverage parsers based on the Wall Street journal to predict psycholinguistic measures of processing difficulty.

Appendix A

Action and Language
Mark Steedman
Informatics, University of Edinburgh

The distinctive character of human information-processing and communication is its context-dependence. While cognition is compositional, in the sense that thoughts are functions of their parts, the compositional process itself is hugely ambiguous. Most of the problem of information processing is that of eliminating ambiguity and (equivalently) supporting efficient inference.

For example, representations in the visual cortex are map-like in relation to the retina. We know that this is not a mere accident of developmental inertia, because the auditory cortex of nocturnal animals like owls is also map-like, although the original sense-data is not. In both cases, the cortical represnetation is map-like because it supports efficient inference about spatial relatiions that are relevant to acting in the world.

The logic-inspired computational formalisms for supporting world knowledge generally fail to support efficient inference for realistic domains. Recent interest in an alternative brain-inspired information and communication technology hopes to discover knowledge representations and semantic forms that do better, by being directly grounded in our way of being in the world in the way that the owl's auditory cortex is, and by being probabilistic and associative, rather than purely logical.

Since the work of Lashley and Miller, Galanter and Pribram in the '50s, it has been commonplace to associate language and other distinctively human aspects of serial behavior with an underlying more primitive ability to make plans and reason about actions that we share with our closest animal relatives, and to seek a common neural substrate for those faculties and sensory-motor planning. Recently this link has been reinforced by the observation of co-localization of brain activity caused by verbs and the motor actions they denote (Pulvermüller 2002). Behavioural experiments have further demonstrated a bidirectional interaction between processing of actions, and processing of words related to those actions (Pulvermüller 2005; Boulenger et al. 2006; Scorolli and Borghi 2007; Glenberg, Sato and Cattaneo 2008.

It is interesting to ask whether the techniques that are used for planning robot action can be applied to the analysis of this underlying sensory motor system, in order to see more clearly the way in which language and other related cognitive ability could have become attached to it in what in evolutionary terms amounts to a mere instant of a few million years.

The term planning is used in two distinct senses in the AI literature. One sense is used at the level of motor-control, where it refers to the problem of finding optimal paths through state spaces defined by often large amounts of raw sensory-motor data, using techniques like dynamic programming (Bellman 1957; Sutton 1991). The other sense of the term refers to the problem of finding sequences of actions to reach a state that satisfies some goal. States in this second search space are abstract descriptions, represented by vectors of symbolic, essentially propositional, features, representing properties of grounded named entities such as $door(d1)$ and $open(d1)$. For reasons of speed and efficiency, rules of this second kind, representing actions in such state-spaces, are represented by underspecified vectors, capturing changes in just a few features of this kind, as in STRIPS rules (Fikes and Nilsson 1971), such as the following, which means "if a *door* is *shut* and you *push it*, it stops being shut and becomes *open*":

$$\{door(x)\}, shut(x) - \circ [push(x)] open(x)$$

For similar reasons, such rules are often "deictic" in nature, applying to a small portion of the world-state defined by a focus of attention (Agre and Chapman 1987; Duchon, Warren and Kaelbling 1998).

Features like $door(d1)$ and actions like $push(d1)$ are essentially *structured*—that is, composed of a predicate and one or more arguments—while expressions like $door(x)$ and $push(x)$ in STRIPS rules like the above implicitly existentially quantify (or, equivalently, abstract) over argument positions in such structured meanings.

It is hard to imagine how an animal could make plans of the kind investigated by Köhler (1925), involving iterated use of tools such as (arbitrary numbers of) boxes to attain goals such as obtaining bananas, without such structured representations and some means of abstracting over individual instances.

It is also clear from Koehler's work that such planning in apes is "object-oriented", depending on the immediate perceptual availability of the necessary tool, suggesting that a notion of "affordance" is involved, associating objects like boxes with the actions that they mediate. Steedman (2002b,a, 2004) associates affordance of objects with type-raising and seriation of affordances into plans of action with function composition, and points out that these two combinatory operators can be viewed as fundamental to natural language syntax.

If such planning is "deliberative", rather than purely reactive and policy-driven, and involves searching a disjunctive branching space (or "Kripke model") of alternative composed sequences of actions under some regime such as forward-chaining breadth-first or iterative-deepening search, it requires a push-down automaton (or more likely a finite-state simulation of a PDA) to keep track of intermediate states and evaluations.

However, while apes (and presumably the common ancestor that we share with them) appear to have access to affordance and seriation, and to define states in terms of structured meanings, the apes, at least, do not appear to have truly *recursive* predicates in those structured meanings(Premack and Premack 1983; Premack 1986). Propositional attitude predicates like (someone) *knowing* (some fact) are intrinsically recursive. For example, the following (simplified) STRIPS rule for telling someone something has among its results, not only that they know that thing, but that they know that the speaker knows that thing:

$$\{person(x), \neg know(x,y)\} - \circ [tell(x,y,z)] know(x,y), know(x, know(z,y))$$

(This rule has the useful effect of preempting the possibility that the hearer x might tell the speaker z the same thing back.)

In order to draw the correct inference from recursive structured meanings like $know(x, know(z,y))$, one needs a PDA. It therefore seems likely that such recursion requires the prior evolution of the PDA (or a finite-state simulation thereof) for some independent purpose, such as search for plans.

In this connection it is interesting that, if the ability to plan is extended to recursive actions involving multiple agents, like $make(harry, (make(rabbit, run)))$, in rules like the following, then a generalization of a PDA, the Embedded PDA (EPDA) is required.

$$\{person(x), rabbit(y), \neg running(y)\} - \circ [make(x, (make(y, run)))] running(y)$$

The EPDA is a PDA whose stack can itself contain stack-valued elements. The reason this generalization is required is that such plans can build up an (in principle) unlimited number of participants, represented here by variables $x, y, \ldots$ (although since the EPDA must also be simulated in the same sense as the PDA, there will be a bound on their number). Vijay-Shanker and Weir (1994) show that the EPDA is characteristic of some low-power generalizations of CFG that have been claimed to be strongly adequate for capturing natural language, notably Tree adjoining grammar (TAG, Joshi 1988) and Combinatory Categorial Grammar (CCG, Steedman 2000b). In particular, the latter framework requires only the addition of combinatory operations of type-raising and composition to project lexical concepts onto syntactic derivations. We have seen that such operations are already implicit in the mammalian planning system. Koechlin and Jubault (2006) present fMRI evidence for a model of prefrontal cortex for processing start and end points of functional segments of hierarchical action plans, which they argue would support processing structures with multiple hierarchical levels.

It seems likely that it is the availability of recursive concepts of mind related to the ability to make collaborative plans of the kind discussed by Hrdy (2009) that distinguishes humans from their closest anthropoid cousins, and distinctively supports natural language. (Indeed, it is hard to see how either coperative planning or communicative language could exist without the support of such necessarily recursive propositional attitude concepts.)

We may therefore hypothesize that this conceptual-semantic understanding of mind is what underlies the recursivity property of language and other distinctively cognitive abilities, including music and even reflective consciousness itself, rather than the distinctively syntactic source postulated by Hauser, Chomsky and Fitch (2002)

In summary, we may conjecture that the following progression, spanning a couple of hundred million years of evolution of the mammalian neocortex, provided the necessary substrate for the essentially instantaneous subsequent development of human language and the other cognitive faculties we have mentioned.

1. Reactive planning with a nonrecursive KR (finite state)
2. Deliberative planning with a nonrecursive KR using (forward) chaining and breadth-first fixed-depth or iterative deepening requires a (simulated) PDA, as well as composition and type-raising.
3. A PDA supports recursion in the KR language.
4. Plan Inference in a recursive KR language requires simulating an EPDA.
5. An EPDA supports attested NL grammar

It is important to realize that we cannot expect to recapitulate this entire evolutionary progression using machine learning over a *tabula rasa* operating in a universe of pure sense data. Evolution operates with essentially unbounded resources of computational space and time and a disregard for consequences that it is inconceivable we can ever afford.

However, we can hope to replicate the progression from systems with recursive concepts, composition and type-raising, and a (simulated) EPDA to systems like ourselves, with language and other cognitive faculties that language supports, such as music.

Finney et al. (2002); Zettlemoyer, Pasula and Kaelbling (2005); Modayil and Kuipers (2007b,a) and Mourão, Petrick and Steedman (2008, 2009, 2010), show that STRIPS rules can

be learned from structured propositional state change representations derived automatically from robots and/or simulations of agents acting in the world, using a variety of machine learning techniques such as Hopfield nets, , and kernel-perceptron-based associative networks. Such learning is resistant to noise and partial or incomplete observation, and rules can be extracted in symbolic form from the neurocomputational matrix using techniques under development by Mourão, Petrick and Steedman (2009), and applied using standard planners including the one developed by UEDIN under the FP6 IP PACO-PLUS. This planner is adapted to planning with sensing actions including speech acts such as questioning and informing (Steedman and Petrick 2007).

One may speculate that the grounded structured meaning representations of states and state-changes, including knowledge states and state-changes, that have been induced in such work can in turn be translated into a structured semantic representation or language of logical form that will provides a suitable substrate for acquisition of a categorial lexicon for a range of grammar fragments in a number of natural languages.

A crucial element of such meaning representations that has been missing from previous computational studies of language acquisition, such as Siskind (1995); Villavicencio (2002); Zettlemoyer and Collins (2005); Lu et al. (2008), is the interpersonal dimension—that "more cookies" not only denotes cookies, but an act of offering or shared attention. In English, markers of shared attention and propositional attitude including information-structural notions of "topic", "comment", and novelty, are signaled by intonation and prosody (Steedman 2000a). Very young children show exquisite sensitivity to these aspects of utterance, long before they control other aspects of language (Fernald 1993). Steedman (2000a) shows that such information-structural aspects of meaning are isomorphic to syntactic and semantic derivation, once the EPDA is defined as the characteristic automaton for both planning and natural language processing.

It remains to show that the EPDA model immediately generalizes to robust planning and NLP, and that standard statistical optimization techniques that we have already applied to planning and CCG parsing (Petrick and Bacchus 2004; Clark and Curran 2004) can be applied to make our theory scale to large planning domains, grammars, and parsing models that are required for practical applications.

Such a demonstration will provide proof of concept for a relation of action and language, and a role for probabilistic models in processing, that will bring the linguistic meaning representation much closer to efficient grounded knowledge representation, to support tasks like real-world inference, textual entailment for question answering, and statistical MT that existing representations, based on standard predicate logics and first-order theorem provers, typically fail on badly.

References

Agre, Phillip, and David Chapman. 1987. "Pengi: An Implementation of a Theory of Activity." In *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*. Los Altos, CA: Morgan Kaufmann.

Bellman, Richard. 1957. *Dynamic Programming*. Princeton NJ: Princeton University Press.

Boulenger, Veronique, Alice C. Roy, Yves Paulignan, Viviane Deprez, Marc Jeannerod, and Tatjana A. Nazir. 2006. "Cross-tAlk between Language Processes and Overt Motor Behavior in the First 200 msec of Processing." *Journal of Cognitive Neuroscience*, 18, 10, 1607–1615.

Clark, Stephen, and James R. Curran. 2004. "Parsing the WSJ using CCG and Log-Linear Models." In *Proceedings of the 42nd Meeting of the ACL*, 104–111. Barcelona, Spain.

Duchon, A., W. Warren, and Leslie Kaelbling. 1998. "Ecological Robotics." *Adaptive Behavior*, 6, 473–507.

Fernald, Anne. 1993. "Approval and Disapproval: Infant Responsiveness to Vocal Affect in Familiar and Unfamiliar Languages." *Child Development*, 64, 657–667.

Fikes, Richard, and Nils Nilsson. 1971. "STRIPS: a New Approach to the Application of Theorem Proving to Problem Solving." *Artificial Intelligence*, 2, 189–208.

Finney, Sarah, Natalia H. Gardiol, Leslie Pack Kaelbling, and Tim Oates. 2002. "Learning with Deictic Representation." AI Laboratory Memo AIM-2002-006, MIT.

Glenberg, A., M. Sato, and L. Cattaneo. 2008. "Use-Induced Motor Plasticity Affects the Processing of Abstract and Concrete Language." *Current Biology*, 18, 7, R290–R291.

Hauser, Marc, Noam Chomsky, and W. Tecumseh Fitch. 2002. "The Faculty of Language: What Is It, Who Has It, and How did it Evolve?" *Science*, 298, 1569–1579.

Hrdy, Sarah Blaffer. 2009. *Mothers and Others*. Cambridge, MA: Belnap/Harvard University Press.

Joshi, Aravind. 1988. "Tree Adjoining Grammars." In David Dowty, Lauri Karttunen, and Arnold Zwicky, eds., *Natural Language Parsing*, 206–250. Cambridge: Cambridge University Press.

Koechlin, Etienne, and Thomas Jubault. 2006. "Broca's Area and the Hierarchical Organization of Human Behavior." *Neuron*, 50.

Köhler, Wolfgang. 1925. *The Mentality of Apes*. New York: Harcourt Brace and World.

Lashley, Karl. 1951. "The Problem of Serial Order in Behavior." In L.A. Jeffress, ed., *Cerebral Mechanisms in Behavior*, 112–136. New York: Wiley. Reprinted in Saporta (1961).

Lu, Wei, Hwee Tou Ng, Wee Sun Lee, and Luke S. Zettlemoyer. 2008. "A Generative Model for Parsing Natural Language to Meaning Representations." In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 783–792. Association for Computational Linguistics.

Miller, George, Eugene Galanter, and Karl Pribram. 1960. *Plans and the Structure of Behavior*. New York, NY: Henry Holt.

Modayil, Joseph, and Ben Kuipers. 2007a. "Autonomous development of a grounded object ontology by a learning robot." In *Proceedings of the AAAI Spring Symposium on Control Mechanisms for Spatial Knowledge Processing in Cognitive/Intelligent Systems*. AAAI.

Modayil, Joseph, and Ben Kuipers. 2007b. "Where Do Actions Come From? Autonomous Robot Learning of Objects and Actions." In *Proceedings of the AAAI Spring Symposium on Control Mechanisms for Spatial Knowledge Processing in Cognitive/Intelligent Systems*. AAAI.

Mourão, Kira, Ron Petrick, and Mark Steedman. 2008. "Using Kernel Perceptrons to Learn Action Effects for Planning." In *Proceedings of 3rd International Conference on Cognitive Systems (CogSys 2008)*, 45–50. University of Karlsruhe. URL `http://www.cogsys2008.org`.

Mourão, Kira, Ron Petrick, and Mark Steedman. 2009. "Learning Action Effects in Partially Observable Domains." In *Proceedings of the ICAPS 2009 Workshop on Planning and Learning*, 15–22. Thessaloniki, Greece.

Mourão, Kira, Ron Petrick, and Mark Steedman. 2010. "Learning Action Effects in Partially Observable Domains." In *Proceedings of the 19th European Conference on AI*. Lisbon.

Petrick, Ronald P. A., and Fahiem Bacchus. 2004. "Extending the Knowledge-Based Approach to Planning with Incomplete Information and Sensing." In Shlomo Zilberstein, Jana Koehler, and Sven Koenig, eds., *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS-04)*, 2–11. Menlo Park, CA: AAAI Press.

Premack, David. 1986. *Gavagai!*. Cambridge, MA: MIT Press, Bradford Books.

Premack, David, and Ann James Premack. 1983. *The Mind of an Ape*. New York, NY: Norton.

Pulvermüller, Friedemann. 2002. *The Neuroscience of Language*. Cambridge University Press.

Pulvermüller, Friedemann. 2005. "Brain Mechanisms Linking Language and Action." *Nature Reviews Neuroscience*, 6, 7, 576–582.

Saporta, Sol, ed. 1961. *Psycholinguistics: A Book of Readings*. New York: Holt Rinehart Winston.

Scorolli, Claudia, and Anna M. Borghi. 2007. "Sentence Comprehension and Action: Effector Specific Modulation of the Motor System." *Brain Research*, 1130, 1, 119–124.

Siskind, Jeffrey. 1995. "Grounding Language in Perception." *Artificial Intelligence Review*, 8, 371–391.

Steedman, Mark. 2000a. "Information Structure and the Syntax-Phonology Interface." *Linguistic Inquiry*, 34, 649–689.

Steedman, Mark. 2000b. *The Syntactic Process*. Cambridge, MA: MIT Press.

Steedman, Mark. 2002a. "Formalizing Affordance." In *Proceedings of the 24th Annual Meeting of the Cognitive Science Society, Fairfax VA, August*, 834–839. Mahwah NJ: Lawrence Erlbaum.

Steedman, Mark. 2002b. "Plans, Affordances, and Combinatory Grammar." *Linguistics and Philosophy*, 25, 723–753.

Steedman, Mark. 2004. "Where Does Compositionality Come From?" In Simon Levy and Ross Gaylor, eds., *Proceedings of the AAAI Fall Symposium on Compositional Connectionism in Cognitive Science*, 59–62. Menlo Park: AAAI. Technical Report FS-04-03.

Steedman, Mark, and Ron Petrick. 2007. "Planning Dialog Actions." In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, 265–272. Antwerp, Sept., ACL.

Sutton, Richard. 1991. "Planning by Incremental Dynamic Programming." In *Proceedings of the 9th Conference on Machine Learning*, 353–357. San Francisco: Morgan Kauffman.

Vijay-Shanker, K., and David Weir. 1994. "The Equivalence of Four Extensions of Context-Free Grammar." *Mathematical Systems Theory*, 27, 511–546.

Villavicencio, Aline. 2002. *The Acquisition of a Unification-Based Generalised Categorial Grammar*. Ph.D. thesis, University of Cambridge.

Zettlemoyer, Luke, and Michael Collins. 2005. "Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars." In *Proceedings of the 21st Conference on Uncertainty in AI (UAI)*, 658–666. ACL.

Zettlemoyer, Luke S., Hanna M. Pasula, and Leslie Pack Kaelbling. 2005. "Learning Planning Rules in Noisy Stochastic Worlds." In *National Conference on Artificial Intelligence (AAAI),*. AAAI.

Appendix B

Computational Grammar Acquisition from CHILDES data using a Probabilistic Parsing Model.

Tom Kwiatkowski, Sharon Goldwater, Mark Steedman
School of Informatics, University of Edinburgh
t.m.kwiatkowski@sms.ed.ac.uk,{sgwater, steedman}@inf.ed.ac.uk

1 Introduction

In this work we propose a universal model of syntactic acquisition that assumes the learner is exposed to pairs consisting of strings of word-candidates and contextually-afforded meaning-representations.

Previous attempts to model the learning of syntax (Siskind 1992, 1995, 1996; Villavicencio 2002; Yang 2002; Buttery 2003) have tended to adopt a "parameter-setting" approach (Hyams 1986; Gibson & Wexler 1995; Fodor 1998). However, recent work in the related task of inducing a grammar from a corpus of paired English sentences and database queries (Zettlemoyer & Collins 2005, Zettlemoyer & Collins 2007, Wong & Mooney 2007, Lu et al. 2008) has shown that it is possible to learn grammars without this "switch like" mechanism by using the structure of the meaning representation to bootstrap the syntactic learning procedure.

The present paper shows that these related methods can be generalized to provide a universal model of child language acquisition and our model is designed to be psycholinguistically plausible: the initialisation of the grammar is language independent and should be able to learn any plausible word order; and the model learns in a sequential manner from sentence - meaning pairs. For the purposes of this paper, we present only the case of learning from unambiguous sentence-meaning pairs. However, the principles used will extend to the case of learning in the face of spurious distracting meaning candidates that are contextually supported but irrelevant to the utterance.

2 Logical Form

Sagae et al. 2007 have recently annotated a substantial part of the English section of the CHILDES database with dependency graphs of the kind illustrated in figure 1. While this annotation scheme was designed to represent syntactic relations, these dependency graphs can be viewed as impoverished logical forms representing pure predicate-argument meaning relations, provided that the following language-specific aspects of the annotation are ignored by the learner. First, the learner must make no use of the fact that the dependency graph aligns the terminals of the predicate argument structure with words of English in an English sentence. For example, the learner must consider the possibility that the unknown word "blocks" corresponds to the semantic predicate *get* in figure 2.

Second, the learner must also ignore the fact that the mapping from nodes in the dependency graph to English words is one-to-one. For example, it should consider the possibility that the word "get" corresponds to the compound meaning abbreviated as *get out*.

Third, the learner must map dependency graphs like figure 1 onto structured logical forms like figure 2, in which terms must first be distinguished as functors, arguments, or adjuncts, so that they can be semantically typed.

We can assume that POS tags like *NN*, *VP* and directional dependencies labeled with relations like *jct* in dependency graphs like (1) can be mapped by rule in this way onto semantic types which for mnemonic reasons and ease of reading we will represent as basic unlinearized category schemata *S*, *NP*, *S|NP*, etc.: These type-schemata should be thought of as primarily semantic in nature, distinct from directional syntactic types like *S*, *NP*, $S\backslash NP$, etc. that instantiate them for a particular language The full set of such type schemata is given in figure A-1.

3 CCG Universal Grammar

A Combinatory Categorial Grammar consists of a language-specific lexicon whose entries are triples $\langle word := syntactic\ category : logical\ form \rangle$, and a universal set of syntactic combinatory rules that project the lexicon of a language onto its sentences.

For example, the English lexicon includes the following entries:

the	:=	NP/N	:	*the*
blocks	:=	N	:	*blocks*

The syntactic type NP/N identifies English "the" as combining with nouns of type N to its right to yield NPs. The corresponding lexical entry in a determiner-final language such as Lakhota would be written ki:= $NP\backslash N$: *the*. The logical form *the* is a place-holder for the presumed universal semantics of definites, which may or may not be separately lexicalized in any given language.

The present paper uses only the rules of Application and Harmonic Composition, illustrated in figure A-2 as a result of which, the present system can only learn languages that are weakly context-free. However, it will generalize to the trans-context-free set covered by full CCG.

Consider the case in which a child equipped with the above universal rules but with no lexicon at all hears the sentence MORE DOGGIES! and knows unambiguously that this means *more dogs*. She can ap-

Figure 1: Syntactic dependency graph

$$[_{S_q}\ can_{(S_q|NP_{subj})|(S_{inf}|NP)}]\ [_{S_{inf}|NP}\ get_{(((S_{inf}|NP)|NP)|(S_{nom}|NP))}\ out_{S_{nom}|NP}]\ [_{NP}\ the_{NP|N}\ blocks_N]]\ you_{NP}]$$

Figure 2: pseudo logical form

ply the universal combinatory rules in reverse to the pair *NP* : *more dogs* to directly generate all possible ways that universal rules could project all possible lexical entries, pairing all possible words with all possible decompositions of the logical forms. As the only two combinatory rules that have a non-function category as their result are the rules of function application, the type-and-meaning representation *NP* : *more dogs* generates just three derivations, illustrated in A-1.3.

Of these, the first derivation is correct for determiner-first languages like English. The second would be correct for a determiner-final language like Lakhota. The third would be correct for a language where *more dogs* was realized as a single word.

4 Model

We use a probabilistic parsing model to generate all candidate parses for each sentence/logical-form pair in the training set. This model, described in A-1.4, works by first generating a syntactic parse tree with CCG syntactic categories at the nodes before then generating associated components of logical-form and words at the leaves of this tree. The model makes use of the conjugate-exponential Dirichlet Distribution and Dirichlet Process priors and is trained using the online Variational Bayesian Expectation Maximisation algorithm (Beal (2003)). This training procedure is online in the strong sense that each training pair is seen sequentially and only once.

5 Experiments

The model is trained on a set of 3599 child-directed sentence; logical-form pairs from the first 15 files of the Eve corpus discussed in section 2. These were collected between the ages of 1;6 and 2;1 (years; months) and only those sentences of 6 words or fewer were used, giving 10^4 word candidates for which the universal grammar licenses 2×10^5 distinct $\langle word, meaning, syntactic\ category \rangle$ triples. Our test set is made up of the child-spoken sentence; logical-form pairs from files 14 and 15 of the Eve corpus (collected at age 2;1).

Our evaluation is similar to that used in the semantic parsing literature, where the parsing model is used to predict logical-forms for a test set of sentences. We score these predicted logical-forms against the gold standard logical-forms with which the test sentences are annotated, reporting both exact-match accuracy and partial-match accuracy, where the latter relates to the directed, labelled, dependencies within the logical-forms.

Table 1 gives precision, recall and f-score for both exact-match accuracy and partial match accuracy. Results are reported for the full test set and also for the subset (79%) of the test set which contains only words that were observed in the training set. These results

Words seen in training set				
		Precision	Recall	f-score
exact-match	baseline	100	13.6	23.9
	model	62	36	**45.5**
partial-match	baseline	100	19	31.9
	model	70	74	**71.9**
Full training set				
		Precision	Recall	f-score
exact-match	baseline	100	10	18.2
	model	51	28	**36.2**
partial-match	baseline	100	16.3	28.0
	model	61.9	67.5	**64.6**

Table 1:

show the parsing model significantly outperforming the baseline of memorised seen sentence-meaning pairs indicating an accurate lexicon and grammar. It should be noted that the training data for our model constitutes only a small subset of the child's full linguistic exposure (34 hours over a 7 month period). We expect would perform with a much higher accuracy if it were given a training set of a comparable size to that available to the child.

6 Conclusion

The above account represents the first step in building a universal model of first language acquisition. We have shown that there is a general method for mapping strings of English paired with impoverished meaning representations derived from dependency annotations onto a grammar/parser that builds such knowledge representations, without any English-specific language engineering, and that the parser trained on a subset of the Eve corpus in a psycholinguistically plausible online manner has built a reasonably accurate model of the CCG lexicon and grammar on the basis of a very small amount of data.

A-1 Supporting Material

A-1.1 Type schemata

S$_{[dcl]}$**:** for declarative sentences
S$_{[wh]}$ **:** for wh questions
S$_{[q]}$ **:** for Yes/No questions
S$_{[to]}|$**NP**$_{SUBJ}$**:** for to-infinitives
S$_{[b]}|$**NP**$_{SUBJ}$**:** for bare-infinitives
NP$_{SUBJ}$**:** for subject noun phrases
NP$_{OBJ}$**:** for object noun phrases
NP$_{PRED}$**:** for predicate noun phrases
NP: for noun phrases
N : for nouns
PP: for prepositional phrases

Figure A-1: Semantic type schemata

A-1.2 CCG combinators

Application
$X{:}f(a) \rightarrow X/Y{:}\lambda x.f(x)\ Y{:}a$
$X{:}f(a) \rightarrow X\backslash Y{:}\lambda x.f(x)\ Y{:}a$

Harmonic Composition
$X/Z{:}\lambda x.f(g(x)) \rightarrow X/Y{:}\lambda x.f(x)\ \ Y/Z{:}\lambda x.g(x)$
$X\backslash Z{:}\lambda x.f(g(x)) \rightarrow Y\backslash Z{:}\lambda x.g(x)\ \ X\backslash Y{:}\lambda x.f(x)$

Figure A-2: CCG combinators

A-1.3 Parse forest

The type-and-meaning representation *NP* : *more N* : *dogs* generates just three derivations:

a. $\dfrac{\dfrac{\text{MORE}}{NP/N : more'_{((e,t),e)}} \quad \dfrac{\text{DOGGIES}}{N : dogs'_{(e,t)}}\ !}{NP : more'dogs'_e}>$

b. $\dfrac{\dfrac{\text{MORE}}{N : dogs'_{(e,t)}} \quad \dfrac{\text{DOGGIES}}{NP\backslash N : more'_{((e,t),e)}}\ !}{NP : more'dogs'_e}<$

c. $\dfrac{\text{MORE DOGGIES}\ !}{NP : more'dogs'_e}$

A-1.4 Parsing Model

In order to generate a parse, the model first generates a syntactic parse tree with CCG syntactic categories at the nodes before then generating associated components of logical-form and words at the leaves of this tree.

We denote a single syntactic node in the parse tree as σ, a single node representing a component of logical-form as λ and a single word node as ϕ.

The generative process used to generate a string of words and associated component of logical-form is illustrated in figure A-1.4 (for which we have borrowed elements of the notation of Liang et al. (2007) since - as they point out - there is no convenient way of representing parse trees in the visual language of traditional graphical models).

This process proceeds by first drawing the top node of the parse tree (σ_{top}) from a Multinomial distribution over the atomic syntactic categories. We then build the tree by recursively drawing either a pair of syntactic children ($\langle \sigma_{l(i)}, \sigma_{r(i)} \rangle$) or a lexical item from each syntactic node σ_i in the parse tree.

In order to decide whether to generate a pair of syntactic children or lexical item for each syntactic node σ_i, the model draws a binary *rule-type* variable (t_i) from a Binomial distribution. If this variable licenses a syntactic expansion then the syntactic children of σ_i are drawn from a Multinomial distribution covering all the possible expansions of σ_i according to the universal grammar.

Alternatively, if t_i indicates that σ_i is a leaf node in the parse then a component of logical form λ_i is drawn from a Multinomial conditioned on the category σ_i and a word ϕ_i is then drawn from a Multinomial conditioned on the $\langle \sigma_i, \lambda_i \rangle$ pair.

We define priors for each of the Multinomial distributions used in this procedure and in the generation of a single parse, the parameters of the Multinomials are drawn from these priors (note that the Binomial distribution is a special case of the Multinomial).

For the Multinomial distributions used in producing the top syntactic node of the tree; the syntactic children of each non-terminal syntactic node; and the *rule-type* variables, we assign conjugate-exponential Dirichlet priors.

For the Multinomial distributions used to generate the components of logical-form and the words however, we cannot use the Dirichlet Distribution prior as the full scope of the lexicon cannot be known to the child (and therefore to our model) before the start of the language acquisition procedure. For these distributions we then use the infinitely expandable (but still conjugate-exponential) Dirichlet Process as a prior.

It should be observed that none of the nodes in figure A-1.4 are observed as we do not know the correct segmentation of either the sentence or the sentential logical-form. However, given the syntactic derivation created and the linear order of the leaf nodes, there is a deterministic (probability 1) mapping between the elements $\phi_i, \lambda_i : i = 1...N$ and the observed pair (S,I), we have just chosen not to depict this in figure A-1.4 for reasons of clarity.

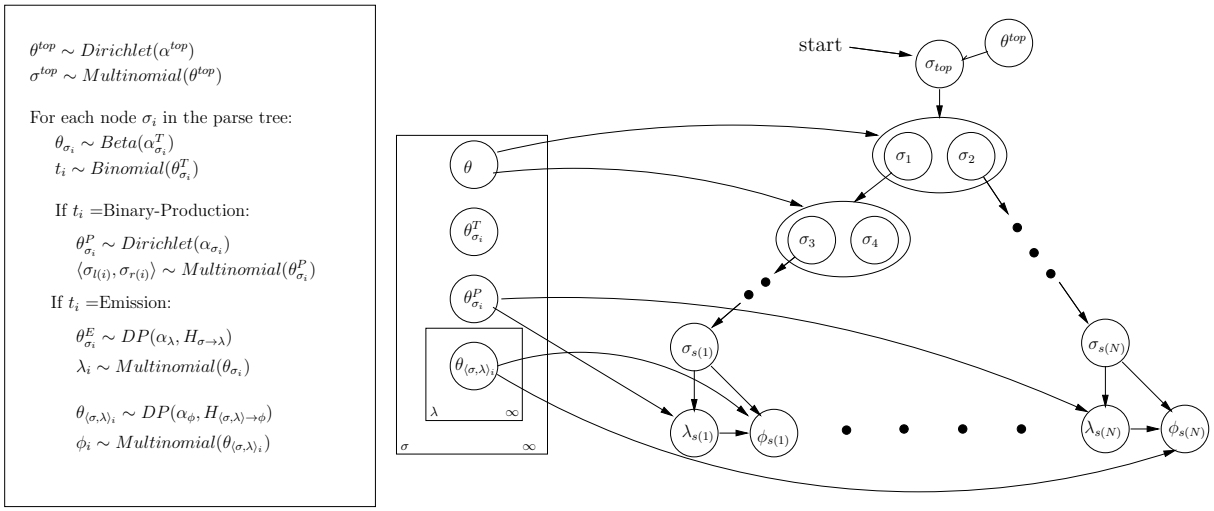


Figure A-3:

References

Beal, M. (2003). Variational algorithms for approximate bayesian inference. Ph.D. thesis, Gatsby Computational Neuroscience Unit, University College London.

Buttery, P. (2003). A computational model of first language acquisition. In *Computational Linguistics UK 6*.

Fodor, J. D. (1998). Unambiguous triggers. *Linguistic Inquiry*, *29*, 1–36.

Gibson, E. & Wexler, K. (1995). Triggers. *Linguistic Inquiry*, *25*, 355–407.

Hyams, N. (1986). *Language Acquisition and the Theory of Parameters*. Dordrecht: Reidel.

Liang, P., Petrov, S., Jordan, M., & Klein, D. (2007). The infinite pcfg using hierarchical dirichlet processes. In *Proc. 2007 Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, East Stroudsburg, PA.

Lu, W., Ng, H. T., Lee, W. S., & Zettlemoyer, L. S. (2008). A generative model for parsing natural language to meaning representations. in proceedings of the conference on empirical methods in natural language processing. In *Proceedings of The Conference on Empirical Methods in Natural Language Processing*.

Sagae, K., Davis, E., Lavie, A., MacWhinney, B., & Wintner, S. (2007). High accuracy annotation and parsing of childes transcripts. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pp. 25–32. held in conjunction with ACL 2007 Prague, ACL.

Siskind, J. (1995). Grounding language in perception. *Artificial Intelligence Review*, *8*, 371–391.

Siskind, J. (1996). A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, *61*, 39–91.

Siskind, J. M. (1992). Naive physics, event perception, lexical semantics, and language acquisition. Ph.D. thesis, MIT.

Villavicencio, A. (2002). The acquisition of a unification-based generalised categorial grammar. Ph.D. thesis, University of Cambridge.

Wong, Y. W. & Mooney, R. (2007). Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pp. 960–967. ACL.

Yang, C. (2002). *Knowledge and Learning in Natural Language*. Oxford: Oxford University Press.

Zettlemoyer, L. & Collins, M. (2005). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the 21st Conference on Uncertainty in AI (UAI)*, pp. 658–666. ACL.

Zettlemoyer, L. & Collins, M. (2007). Online learning of relaxed ccg grammars for parsing to logical form. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL)*, pp. 678–687. ACL.

Appendix C

Probabilistic CCG Grammar Induction using Higher-Order Unification

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, Mark Steedman

Informatics,
University of Edinburgh.

Abstract

1 Introduction

Given the notorious shortage of language-specific syntactically labeled treebank data, there is some interest to the idea that semantic grammars could be induced from naturally-occurring, language-independent corpora of meaning-representations such as database queries, paired with sentences expressing their meaning in the target language.

Recently there has been a variety of work in developing semantic parsers that represent the mapping between sentence and meaning which can be trained on a set of sentences annotated with logical forms (Zettlemoyer & Collins, 2005; Wong & Mooney, 2007; Zettlemoyer & Collins, 2007; Lu et al., 2008). Each of these approaches builds a grammar that maps the words of the sentences to a fragment of such logical-forms then combines these fragments to give a full sentential meaning. Zettlemoyer & Collins (2005, 2007) do this by modeling meanings using a lambda calculus representation and inducing a Combinatory Categorial Grammar (CCG) to control the order of combination of the components of the logical form that are learnt at the level of the lexicon.

CCG is a natural choice for the task of semantic parsing since it is fully transparent to the order of composition of the semantics in the parse tree. Previous work that has used CCG in this way (Zettlemoyer & Collins, 2005, 2007) has relied on the use of language specific syntactic templates to generate the lexical entries required to model the mapping between sentence and meaning.

We introduce a language-independent method of proposing candidate lexical entries by using a constrained form of higher-order unification. This algorithm uses the internal compositional structure of the logical form along with the set of CCG combinators to define a set of syntactic parses for any sentence, logical form pair in a manner that is neither language or domain specific.

We test this approach on the geo250 dataset generally used for evaluation of multilingual semantic parsers and show that we achieve results that are in line with state of the art systems that require significant hand engineered domain specific knowledge.

2 Background.

2.1 Logical Forms.

We follow Zettlemoyer & Collins (2005) in representing the semantics of the sentences in our training data using a typed lambda-calculus language with three basic types: e for entities, t for truth values and i for numbers. These basic types may be combined to form functional types such as $\langle e, t\rangle$, the type of a function that maps from entities to truth values. These functional types may, in turn, be combined to form functional types of arbitrary complexity.

The lambda calculus expressions corresponding to logical forms of the sentences in the training corpus are made of **constants** (which may be entities, numbers or functions); **logical connectors**; **quantifiers** and **lambda bindings** (which identify the variables over which functions apply).

2.2 Higher order unification.

The Higher Order Unification algorithm (Huet, 1975) uses an inversion of the function application and function composition operations on a logical-form **H** to give the set of logical-form pairs {**F, G** } that can be combined to yield **H**. This is summarised below:

Given H, find F,G s.t.

$$\mathbf{H} = F(G)$$

$$\mathbf{H} = \lambda x.F(G(x))$$

In its full form the problem of Higher Order Unification is undecidable but we avoid this complication by constraining the algorithm as described in section.3.1.

2.3 Combinatory Categorial Grammar.

The Combinatory Categorial grammar Steedman (2000) is a strongly lexicalised grammar formalism that couples syntax and semantics tightly, and computes the syntactic derivation and semantic interpretation of a sentence *synchronously*. CCG syntactic categories may be either atomic S, NP or complex A/B, $A\backslash B$ where A and B can themselves be complex. CCG assumes a (functional) mapping between such syntactic categories and the semantic type of the corresponding logical form. For the purposes of the present paper, we assume a very coarse grained semantic type system the number of atomic syntactic categories supported is restricted to NP for anything of type e or i and S for anything of type t.

The slashes in complex categories define the ways in which these categories can apply or compose with adjacent ones in order to form a parse. These slashes are a direct syntactic counterpart of the lambdas used to control variable binding in the logical forms, as illustrated in Fig. 2 which shows the function application and harmonic function composition rules of CCG along with their semantic semantic counterparts. The syntactic categories add information about language-specific word-order—that is, whether each argument is to the left \ or right / of the function.

Application
X/Y:$\lambda x.f(x)$ Y:a → X:$f(a)$
X\Y:$\lambda x.f(x)$ Y:a → X:$f(a)$

Harmonic Composition
X/Y:$\lambda x.f(x)$ Y/Z:$\lambda x.g(x)$ → X/Z:$\lambda x.f(g(x))$
Y\Z:$\lambda x.g(x)$ X\Y:$\lambda x.f(x)$ → X \ Z:$\lambda x.f(g(x))$

Figure 2: CCG combinators

Fig. 1 shows a pair of semantically annotated CCG parses. Each node in the parse tree is signed with a wordspan, logical-form and syntactic category triple. The CCG lexicon Λ stores sets of these triples that can then be used to form parses.

The combinators of Fig. 2 can be inverted to nondeterministically build trees from the top down as is illustrated in Fig. 3

Inverted Application
X:$f(a)$ → X/Y:$\lambda x.f(x)$ Y:a
X:$f(a)$ → X\Y:$\lambda x.f(x)$ Y:a

Inverted Harmonic Composition
X/Z:$\lambda x.f(g(x))$ → X/Y:$\lambda x.f(x)$ Y/Z:$\lambda x.g(x)$
X\Z:$\lambda x.f(g(x))$ → Y\Z:$\lambda x.g(x)$ X\Y:$\lambda x.f(x)$

Figure 3: CCG combinators

This is the syntactic counterpart to the Higher Order Unification algorithm described in section.3.1.

2.4 Log linear models of CCG.

In higher order unification and the rules of the grammar we have a system that defines a set of parses for each logical form, sentence pair. We now introduce a log linear model that can be used to score each of these parses according to a learnt parameter set both at training and test time.

The model consists of a feature vector $\mathbf{f}$ and a parameter vector $\mathbf{w}$ where there is one parameter assigned to each feature. We use $\mathbf{f}(L, T, S)$ to represent the feature set of a logical-form (L); sentence (S); parse tree (T); triple. This triple is then scored using the distribution below.

$$P(L, S, T) = e^{\mathbf{f}(L,T,S).\mathbf{w}}$$

The conditional probabilities $P(|L, S)$, needed at training time and $P(L, T|S)$, needed at test time are calculated as:

$$P(T|L, S) = \frac{e^{\mathbf{f}(\mathbf{L},\mathbf{T},\mathbf{S})\dot{\mathbf{w}}}}{\sum_T e^{\mathbf{f}(\mathbf{L},\mathbf{T},\mathbf{S})\dot{\mathbf{w}}}}$$

$$P(lf, t|s) = \frac{e^{\mathbf{f}(\mathbf{L},\mathbf{T},\mathbf{S})\dot{\mathbf{w}}}}{\sum_{(T,L)} e^{\mathbf{f}(\mathbf{L},\mathbf{T},\mathbf{S})\dot{\mathbf{w}}}}$$

Each of the lexical entries in the grammar Λ has one feature in the feature vector $\mathbf{f}$.

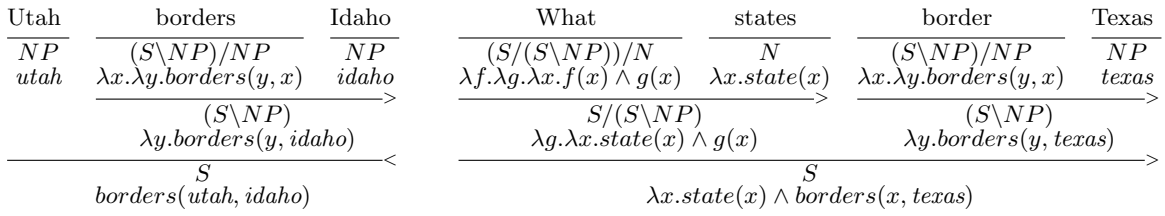


Figure 1: Two examples of CCG parses.

$\mathbf{f}(L, T, S)$ represent the set of lexical features that are used in $\langle L, T, S\rangle$ - one for each lexical item used.

Any configuration of the weight vector $\mathbf{w}$ along with the lexicon Λ represents a probabilistic CCG. Since our grammar framework supports too many lexical items for us to realistically model, we use a sparse representation of the Λ that only has entries for those lexical items which have been used at some point during training. The learning algorithm outlined in the next section that fills this lexicon with entries while also optimising the parameter set $\mathbf{w}$.

3 Learning algorithm

We extend the perceptron learning algorithm of Zettlemoyer & Collins (2007) to use our new language-independent-parse-generation step to propose new lexical candidates where Zettlemoyer & Collins (2007) used language specific templates. The learning algorithm is described in Fig. 4 and only differs in the GENLEX step where we use a constrained form of higher order unification paired with the inverted CCG combinators of 3 to propose new candidate lexical entries.

3.1 Constrained Higher Order Unification.

The problem of Higher Order Unification is, in its full form, undecidable. Here we present a set of constraints that do not impinge upon the grammar's ability to model the dataset used while reducing to a manageable size the number of splits to be considered at each step of the parse generation.

- Force every node in parse to represent a non-empty span of the sentence. This upper bounds the number of nodes in the parse tree.

Inputs: Training examples $\{(x_i, z_i) : i = 1 \ldots n\}$ where each x_i is a sentence, each z_i is a logical form. An initial lexicon Λ_0. Number of training iterations, T.

Definitions: GENLEX(x, z) takes as input a sentence x and a logical form z and returns a set of lexical items as described in section 3.2. GEN$(x; \Lambda)$ is the set of all parses for x with lexicon Λ. GEN$(x, z; \Lambda)$ is the set of all parses for x with lexicon Λ, which have logical form z. The function $\mathbf{f}(x, y)$ represents the features described in section 2.4. The function $L(y)$ maps a parse tree y to its associated logical form.

Initialization: Set parameters $\mathbf{w}$ to 0.1. Set $\Lambda = \Lambda_0$.

Algorithm:

- For $t = 1 \ldots T, i = 1 \ldots n$:

Step 1: (Check correctness)

- Let $y^* = \arg\max_{y \in \text{GEN}(x_i;\Lambda)} \mathbf{w} \cdot \mathbf{f}(x_i, y)$.
- If $L(y^*) = z_i$, go to the next example.

Step 2: (Lexical generation)

- Set $\lambda = \Lambda \cup \text{GENLEX}(x_i, z_i)$.
- Let $y^* = \arg\max_{y \in \text{GEN}(x_i, z_i;\lambda)} \mathbf{w} \cdot \mathbf{f}(x_i, y)$.
- Define λ_i to be the set of lexical entries in y^*.
- Set lexicon to $\Lambda = \Lambda \cup \lambda_i$.

Step 3: (Update parameters)

- Let $y' = \arg\max_{y \in \text{GEN}(x_i;\Lambda)} \mathbf{w} \cdot \mathbf{f}(x_i, y)$.
- If $L(y') \neq z_i$:
  - Set $\mathbf{w} = \mathbf{w} + \mathbf{f}(x_i, y^*) - \mathbf{f}(x_i, y')$.

Output: Lexicon Λ together with parameters $\mathbf{w}$.

Figure 4: An online learning algorithm.

Model	English	Spanish	Turkish	Japanese
WASP	0.70	0.72	0.62	0.74
Lu et al. (2008)	0.72	0.79	0.67	0.76
HUBL	0.79	0.78	0.66	0.76

Table 1: Recall on geo250

- Rule out vacuous extractions of the type $\lambda x.f \;\; a \rightarrow f$.

- Restrict the splits so that they only introduce one new variable. The effect of this restriction is illustrated in Fig. 3.1.

Allow

$$\lambda x f(g(a,x)) \rightarrow \lambda y \lambda x.f(y(x)) \quad \lambda z.g(a,z)$$

Dissallow

$$\lambda x f(g(a,x)) \rightarrow \lambda y \lambda x.f(y(a,x)) \quad \lambda u \lambda z.g(u,z)$$

Figure 5:

3.2 GENLEX

The GENLEX step of the learning algorithm packs the parse chart by sampling a set of 100 parses that correctly map the sentence onto the logical-form. Each of these samples proceeds by first fixing the root node of the parse tree to represent the correct logical-form and covering the full sentence span then recursively splitting nodes in the incomplete parse tree and adding the pair of child nodes to the parse chart.

Each split is performed by first choosing a split of the logical form along with a split of the word span and then assigning each side of this split a syntactic category that is consistent with both the type of the logical form and the syntactic combinator that has been used in reverse to effect the split. We continue to split nodes in the parse tree until each leaf node accounts for a span of one word but we also add all intermediate nodes to the parse chart as candidate lexical entries so that the algorithm is able to learn multiword elements.

4 Experiments

We perform our experiments on the the Geo250 domain of geographical database queries Wong & Mooney (2007). This has natural language sentences in English, Spanish, Turkish and Japanese paired with logical forms.

Evaluation is done using 10-fold cross validation with the same splits used by Wong & Mooney (2007) and Lu et al. (2008). We follow Zettlemoyer & Collins (2005) in initialising the weight vector with a weight of 50 for lexical entries representing each of the named entities supported by the GeoQuery database domain.

We score the algorithm by using it to predict the highest scoring logical form for each of the test sentences and checking this predicted logical form against the gold standard annotation. We report Recall - the ratio of the number of logical-forms correctly predicted to the total number of test sentences - and compare our higher order unification based learner (HUBL) to previous work from WASP Wong & Mooney (2007) and Lu et al. (2008). Our results are directly comparable to those of the WASP system as it uses prolog queries to represent meaning which are similar in form to the λ-calculus used by HUBL. The model of Lu et al. (2008) uses a form of variable free logic to represent the meaning of the sentence which is significantly less expressive than the lambda-calculus used by HUBL so the outputs of these two systems are not directly comparable.

It is apparent from Table. 3 that HUBL achieves higher recall than WASP despite the fact that WASP has access to a significant amount of hard coded domain specific knowledge controlling the allowable composition of the logical forms. HUBL also achieves recalls in line with Lu et al. (2008) despite having to learn from the far bigger hypothesis space

licensed by the lambda-calculus logical forms.

5 Conclusions.

6 Acknowledgements.

Kwiatkowski and Steedman were supported by EU IST Cognitive Systems IP FP6-2004-IST-4-27657 "Paco-Plus" and an EPRSC studentship. Zettlemoyer was supported by a US NSF International Research Fellowship.

References

Huet, G. (1975). A unification algorithm for typed λ-calculus. *Theoretical Computer Science*, *1*, 27–57.

Lu, W., Ng, H. T., Lee, W. S., & Zettlemoyer, L. S. (2008). A generative model for parsing natural language to meaning representations. in proceedings of the conference on empirical methods in natural language processing. In *Proceedings of The Conference on Empirical Methods in Natural Language Processing*.

Steedman, M. (2000). *The Syntactic Process*. Cambridge, MA: MIT Press.

Wong, Y. W. & Mooney, R. (2007). Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pp. 960–967. ACL.

Zettlemoyer, L. & Collins, M. (2005). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the 21st Conference on Uncertainty in AI (UAI)*, pp. 658–666. ACL.

Zettlemoyer, L. & Collins, M. (2007). Online learning of relaxed ccg grammars for parsing to logical form. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL)*, pp. 678–687. ACL.

Appendix D

Unbounded Dependency Recovery for Parser Evaluation

Laura Rimell and **Stephen Clark**
University of Cambridge
Computer Laboratory
laura.rimell@cl.cam.ac.uk
stephen.clark@cl.cam.ac.uk

Mark Steedman
University of Edinburgh
School of Informatics
steedman@inf.ed.ac.uk

Abstract

This paper introduces a new parser evaluation corpus containing around 700 sentences annotated with unbounded dependencies, from seven different grammatical constructions. We run a series of off-the-shelf parsers on the corpus to evaluate how well state-of-the-art parsing technology is able to recover such dependencies. The overall results range from 25% accuracy to 59%. These low scores call into question the validity of using Parseval scores as a general measure of parsing capability. We discuss the importance of parsers being able to recover unbounded dependencies, given their relatively low frequency in corpora. We also analyse the various errors made on these constructions by one of the more successful parsers.

1 Introduction

Statistical parsers are now obtaining Parseval scores of over 90% on the WSJ section of the Penn Treebank (Bod, 2003; Petrov and Klein, 2007; Huang, 2008; Carreras et al., 2008). McClosky et al. (2006) report an F-score of 92.1% using self-training applied to the reranker of Charniak and Johnson (2005). Such scores, in isolation, may suggest that statistical parsing is close to becoming a solved problem, and that further incremental improvements will lead to parsers becoming as accurate as POS taggers.

A single score in isolation can be misleading, however, for a number of reasons. First, the single score is an aggregate over a highly skewed distribution of all constituent types; evaluations which look at individual constituent or dependency types show that the accuracies on some, semantically important, constructions, such as coordination and PP-attachment, are much lower (Collins, 1999). Second, it is well known that the accuracy of parsers trained on the Penn Treebank degrades when they are applied to different genres and domains (Gildea, 2001). Finally, some researchers have argued that the Parseval metrics (Black et al., 1991) are too forgiving with respect to certain errors and that an evaluation based on syntactic dependencies, for which scores are typically lower, is a better test of parser performance (Lin, 1995; Carroll et al., 1998).

In this paper we focus on the first issue, that the performance of parsers on some constructions is much lower than the overall score. The constructions that we focus on are various unbounded dependency constructions. These are interesting for parser evaluation for the following reasons: one, they provide a strong test of the parser's knowledge of the grammar of the language, since many instances of unbounded dependencies are difficult to recover using shallow techniques in which the grammar is only superficially represented; and two, recovering these dependencies is necessary to completely represent the underlying predicate-argument structure of a sentence, useful for applications such as Question Answering and Information Extraction.

To give an example of the sorts of constructions we are considering, and the (in)ability of parsers to recover the corresponding unbounded dependencies, none of the parsers that we have tested were able to recover the dependencies shown in bold from the following sentences:

We have also developed techniques for recognizing and locating underground nuclear tests through the **waves** *in the ground which they* **generate**.

By Monday , they hope to have a sheaf of **documents** *both sides can* **trust**.

By means of charts showing wave-travel times and depths in the ocean at various locations , it is possible to estimate the **rate** *of approach and probable* **time** *of arrival at Hawaii* **of** *a tsunami getting under way at any spot in the Pacific .*

The contributions of this paper are as follows. First, we present the first set of results for the recovery of a variety of unbounded dependencies, for a range of existing parsers. Second, we describe the creation of a publicly available unbounded dependency test suite, and give statistics summarising properties of these dependencies in naturally occurring text. Third, we demonstrate that performing the evaluation is surprisingly difficult, because of different conventions across the parsers as to how the underlying grammar is represented. Fourth, we show that current parsing technology is very poor at representing some important elements of the argument structure of sentences, and argue for a more focused construction-based parser evaluation as a complement to existing grammatical relation-based evaluations. We also perform an error-analysis for one of the more successful parsers.

There has been some prior work on evaluating parsers on long-range dependencies, but no work we are aware of that has the scope and focus of this paper. Clark et al. (2004) evaluated a CCG parser on a small corpus of object extraction cases. Johnson (2002) began the body of work on inserting traces into the output of Penn Treebank (PTB) parsers, followed by Levy and Manning (2004), among others. This PTB work focused heavily on the representation in the Treebank, evaluating against patterns in the trace annotation. In this paper we have tried to be more "formalism-independent" and construction focused.

2 Unbounded Dependency Corpus

2.1 The constructions

An unbounded dependency construction contains a word or phrase which appears to have been moved, while being interpreted in the position of the resulting "gap". An unlimited number of clause boundaries may intervene between the moved element and the gap (hence "unbounded").

The seven constructions in our corpus were chosen for being relatively frequent in text, compared to other unbounded dependency types, and relatively easy to identify. An example of each construction, along with its associated dependencies, is shown in Table 1. Here we give a brief description of each construction.

Object extraction from a relative clause is characterised by a relative pronoun (a *wh*-word or *that*) introducing a clause from which an argument in object position has apparently been extracted: *the paper which I wrote*. Our corpus includes cases where the extracted word is (semantically) the object of a preposition in the verb phrase: *the agency that I applied to*.

Object extraction from a reduced relative clause is essentially the same, except that there is no overt relative pronoun: *the paper I wrote; the agency I applied to*. We did not include participial reduced relatives such as *the paper written by the professor*.

Subject extraction from a relative clause is characterised by the apparent extraction of an argument from subject position: *the instrument that measures depth*. A relative pronoun is obligatory in this construction. Our corpus includes passive subjects: *the instrument which was used by the professor*.

Free relatives contain relative pronouns without antecedents: *I heard what she said*, where *what* does not refer to any other noun in the sentence. Free relatives can usually be paraphrased by noun phrases such as *the thing she said* (a standard diagnostic for distinguishing them from embedded interrogatives like *I wonder what she said*). The majority of sentences in our corpus are object free relatives, but we also included some adverbial free relatives: *She told us how to do it*.

Object *wh*-questions are questions in which the *wh*-word is the semantic object of the verb: *What did you eat?*. Objects of prepositions are included: *What city does she live in?*. Also included are a few cases where the *wh*-word is arguably adverbial, but is selected for by the verb: *Where is the park located?*.

Right node raising (RNR) is characterised by coordinated phrases from which a shared element apparently moves to the right: *Mary saw and Susan bought the book*. This construction is unique within our corpus in that the "raised" element can have a wide variety of grammatical functions. Examples include: noun phrase object of verb, noun phrase object of preposition (*material about or messages from the communicator*), a combination of the two (*applied for and won approval*), prepositional phrase modifier (*president and chief executive of the company*), infinitival modifier (*the will and the capacity to prevent the event*), and modified noun (*a good or a bad decision*).

Subject extraction from an embedded clause is characterised by a semantic subject which is ap-

Object extraction from a relative clause
Each must match Wisman's "pie" with the **fragment** that he **carries** with him.
`dobj(carries, fragment)`
Object extraction from a reduced relative clause
Put another way, the decline in the yield suggests stocks have gotten pretty rich in price relative to the **dividends** they **pay**, some market analysts say.
`dobj(pay, dividends)`
Subject extraction from a relative clause
It consists of a **series** of pipes and a pressure-measuring **chamber** which **record** the rise and fall of the water surface.
`nsubj(record, series)` `nsubj(record, chamber)`
Free relative
He tried to ignore **what** his own common sense **told** him, but it wasn't possible; her motives were too blatant.
`dobj(told, what)`
Object *wh*-question
What **city** does the Tour de France end **in**?
`pobj(in, city)`
Right node raising
For the third year in a row, consumers voted Bill Cosby **first** and James Garner **second in** persuasiveness as spokesmen in TV commercials, according to Video Storyboard Tests, New York.
`prep(first, in)` `prep(second, in)`
Subject extraction from an embedded clause
In assigning to God the **responsibility** which he learned could not **rest** with his doctors, Eisenhower gave evidence of that weakening of the moral intuition which was to characterize his administration in the years to follow.
`nsubj(rest, responsibility)`

Table 1: Examples of the seven constructions in the unbounded dependency corpus.

parently extracted across two clause boundaries, as shown in the following bracketing (where $*$ marks the origin of the extracted element): *the responsibility which [the government said [$*$ lay with the voters]]*. Our corpus includes sentences where the embedded clause is a so-called small clause, i.e. one with a null copula verb: *the plan that she considered foolish*, where *plan* is the semantic subject of *foolish.*

2.2 The data

The corpus consists of approximately 100 sentences for each of the seven constructions; 80 of these were reserved for each construction for testing, giving a test set of 560 sentences in total, and the remainder were used for initial experimentation (for example to ensure that default settings for the various parsers were appropriate for this data). We did not annotate the full sentences, since we are only interested in the unbounded dependencies and full annotation of such a corpus would be extremely time-consuming.

With the exception of the question construction, all sentences were taken from the PTB, with roughly half from the WSJ sections (excluding 2-21 which provided the training data for many

of the parsers in our set) and half from Brown (roughly balanced across the different sections). The questions were taken from the question data in Rimell and Clark (2008), which was obtained from various years of the TREC QA track. We chose to use the PTB as the main source because the use of traces in the PTB annotation provides a starting point for the identification of unbounded dependencies.

Sentences were selected for the corpus by a combination of automatic and manual processes. A regular expression applied to PTB trees, searching for appropriate traces for a particular construction, was first used to extract a set of candidate sentences. All candidates were manually reviewed and, if selected, annotated with one or more grammatical relations representing the relevant unbounded dependencies in the sentence. Some of the annotation in the treebank makes identification of some constructions straightforward; for example right node raising is explicitly represented as RNR. Indeed it may have been possible to fully automate this process with use of the `tgrep` search tool. However, in order to obtain reliable statistics regarding frequency of occurrence, and to ensure a high-quality resource, we used fairly broad regular expressions to identify the original set followed by manual review.

We chose to represent the dependencies as grammatical relations (GRs) since this format seemed best suited to represent the kind of semantic relationship we are interested in. GRs are head-based dependencies that have been suggested as a more appropriate representation for general parser evaluation than phrase-structure trees (Carroll et al., 1998). Table 1 gives examples of how GRs are used to represent the relevant dependencies. The particular GR scheme we used was based on the Stanford scheme (de Marneffe et al., 2006); however, the specific GR scheme is not too crucial since the whole sentence is not being represented in the corpus, only the unbounded dependencies.

3 Experiments

The five parsers described in Section 3.2 were used to parse the test sentences in the corpus, and the percentage of dependencies in the test set recovered by each parser for each construction was calculated. The details of how the parsers were run and how the parser output was matched against the gold standard are given in Section 3.3. This is essentially a recall evaluation, and so is open to abuse; for example, a program which returns all the possible word pairs in a sentence, together with all possible labels, would score 100%. However, this is easily guarded against: we simply assume that each parser is being run in a "standard" mode, and that each parser has already been evaluated on a full corpus of GRs in order to measure precision and recall across all dependency types. (Calculating precision for the unbounded dependency evaluation would be difficult since that would require us to know how many *incorrect* unbounded dependencies were returned by each parser.)

Construction	WSJ	Brown	Overall
Obj rel clause	2.3	1.1	1.4
Obj reduced rel	2.7	2.8	2.8
Sbj rel clause	10.1	5.7	7.4
Free rel	2.6	0.9	1.3
RNR	2.2	0.9	1.2
Sbj embedded	2.0	0.3	0.4

Table 2: Frequency of constructions in the PTB (percentage of sentences).

3.1 Statistics relating to the constructions

Table 2 shows the percentage of sentences in the PTB, from those sections that were examined, which contain an example of each type of unbounded dependency. Perhaps not surprisingly, root subject extractions from relative clauses are by far the most common, with the remaining constructions occurring in roughly between 1 and 2% of sentences. Note that, although examples of each individual construction are relatively rare, the combined total is over 10% (assuming that each construction occurs independently). Section 6 contains a discussion regarding the frequency of occurrence of these events and the consequences of this for parser performance.

Table 3 shows the average and maximum distance between head and dependent for each construction, as measured by the difference between word indices. This is a fairly crude measure of distance but gives some indication of how "long-range" the dependencies are for each construction. The cases of object extraction from a relative clause and subject extraction from an embedded clause provide the longest dependencies, on average. The following sentence gives an example of how far apart the head and dependent can be in a

Construction	Avg Dist	Max Dist
Obj rel clause	6.8	21
Obj reduced rel	3.4	8
Sbj rel clause	4.4	18
Free rel	3.4	16
Obj wh-question	4.8	9
RNR	4.8	23
Sbj embedded	7.0	21

Table 3: Distance between head and dependent.

subject embedded construction:

the same **stump** *which had impaled the car of many a guest in the past thirty years and which he refused to have* **removed**.

3.2 The parsers

The parsers that we chose to evaluate are the C&C CCG parser (Clark and Curran, 2007), the Enju HPSG parser (Miyao and Tsujii, 2005), the RASP parser (Briscoe et al., 2006), the Stanford parser (Klein and Manning, 2003), and the DCU post-processor of PTB parsers (Cahill et al., 2004), based on LFG and applied to the output of the Charniak and Johnson reranking parser. Of course we were unable to evaluate every publicly available parser, but we believe these are representative of current wide-coverage robust parsing technology.[1]

The C&C parser is based on CCGbank (Hockenmaier and Steedman, 2007), a CCG version of the Penn Treebank. It is ideally suited for this evaluation because CCG was designed to capture the unbounded dependencies being considered. The Enju parser was designed with a similar motivation to C&C, and is also based on an automatically extracted grammar derived from the PTB, but the grammar formalism is HPSG rather than CCG. Both parsers produce head-word dependencies reflecting the underlying predicate-argument structure of a sentence, and so in theory should be straightforward to evaluate.

The RASP parser is based on a manually constructed POS tag-sequence grammar, with a statistical parse selection component and a robust partial-parsing technique which allows it to return output for sentences which do not obtain a full spanning analysis according to the grammar. RASP has not been designed to capture many of the dependencies in our corpus; for example, the tag-sequence grammar has no explicit representation of verb subcategorisation, and so may not know that there is a missing object in the case of extraction from a relative clause (though it does recover some of these dependencies). However, RASP is a popular parser used in a number of applications, and it returns dependencies in a suitable format for evaluation, and so we considered it to be an appropriate and useful member of our parser set.

The Stanford parser is representative of a large number of PTB parsers, exemplified by Collins (1997) and Charniak (2000). The Parseval scores reported for the Stanford parser are not the highest in the literature, but are competitive enough for our purposes. The advantage of the Stanford parser is that it returns dependencies in a suitable format for our evaluation. The dependencies are obtained by a set of manually defined rules operating over the phrase-structure trees returned by the parser (de Marneffe et al., 2006). Like RASP, the Stanford parser has not been designed to capture unbounded dependencies; in particular it does not make use of any of the trace information in the PTB. However, we wanted to include a "standard" PTB parser in our set to see which of the unbounded dependency constructions it is able to deal with.

Finally, there is a body of work on inserting trace information into the output of PTB parsers (Johnson, 2002; Levy and Manning, 2004), which is the annotation used in the PTB for representing unbounded dependencies. The work which deals with the PTB representation directly, such as Johnson (2002), is difficult for us to evaluate because it does not produce explicit dependencies. However, the DCU post-processor is ideal because it does produce dependencies in a GR format. It has also obtained competitive scores on general GR evaluation corpora (Cahill et al., 2004).

3.3 Parser evaluation

The parsers were run essentially out-of-the-box when parsing the test sentences. The one exception was C&C, which required some minor adjusting of parameters, as described in the parser documentation, to obtain close to full coverage on the data. In addition, the C&C parser comes with a

[1] One obvious omission is any form of dependency parser (McDonald et al., 2005; Nivre and Scholz, 2004). However, the dependencies returned by these parsers are local, and it would be non-trivial to infer from a series of links whether a long-range dependency had been correctly represented. Also, dependency parsers are not significantly better at recovering head-based dependencies than constituent parsers based on the PTB (McDonald et al., 2005).

	Obj RC	Obj Red	Sbj RC	Free	Obj Q	RNR	Sbj Embed	Total
C&C	**59.3**	62.6	80.0	72.6	(**81.2**) 27.5	**49.4**	22.4	(**59.7**) 53.6
Enju	47.3	**65.9**	**82.1**	**76.2**	32.5	47.1	**32.9**	**54.4**
DCU	23.1	41.8	56.8	46.4	27.5	40.8	5.9	35.7
Rasp	16.5	1.1	53.7	17.9	27.5	34.5	15.3	25.3
Stanford	22.0	1.1	74.7	64.3	**41.2**	45.4	10.6	38.1

Table 4: Parser accuracy on the unbounded dependency corpus; the highest score for each construction is in bold; the figures in brackets for C&C derive from the use of a separate question model.

specially designed question model, and so we applied both this and the standard model to the object *wh*-question cases.

The parser output was evaluated against each dependency in the corpus. Due to the various GR schemes used by the parsers, an exact match on the dependency label could not always be expected. We considered a correctly recovered dependency to be one where the gold-standard head and dependent were correctly identified, and the label was an "acceptable match" to the gold-standard label. To be an acceptable match, the label had to indicate the grammatical function of the extracted element at least to the level of distinguishing active subjects, passive subjects, objects, and adjuncts. For example, we allowed an `obj` (object) relation as a close enough match for `dobj` (direct object) in the corpus, even though `obj` does not distinguish different kinds of objects, but we did not allow generic "relative pronoun" relations that are underspecified for the grammatical role of the extracted element.

The differences in GR schemes were such that we ended up performing a time-consuming largely manual evaluation. We list here some of the key differences that made the evaluation difficult.

In some cases, the parser's set of labels was less fine-grained than the gold standard. For example, RASP represents the direct objects of both verbs and prepositions as `dobj` (direct object), whereas the gold-standard uses `pobj` for the preposition case. We counted the RASP output as correctly matching the gold standard.

In other cases, the label on the dependency containing the gold-standard head and dependent was too underspecified to be acceptable by itself. For example, where the gold-standard relation was `dobj(placed,buckets)`, DCU produced `relmod(buckets,placed)` with a generic "relative modifier" label. However, the correct label could be recovered from elsewhere in the parser output, specifically a combination of `relpro(buckets,which)` and `obj(placed,which)`. In this case we counted the DCU output as correctly matching the gold standard.

In some constructions the Stanford scheme, upon which the gold-standard was based, makes different choices about heads than other schemes. For example, in the the phrase *Honolulu, which is the center of the warning system*, the corpus contains a subject dependency with *center* as the head: `nsubj(center,Honolulu)`. Other schemes, however, treat the auxiliary verb *is* as the head of the dependency, rather than the predicate nominal *center*. As long as the difference in head selection was due solely to the idiosyncracies of the GR schemes involved, we counted the relation as correct.

Finally, the different GR schemes treat coordination differently. In the corpus, coordinated elements are always represented with two dependencies. Thus the phrase *they may half* **see** *and half* **imagine** *the old* **splendor** has two gold-standard dependencies: `dobj(see,splendor)` and `dobj(imagine,splendor)`. If a parser produced only the former dependency, but appeared to have the coordination correct, then we awarded two marks, even though the second dependency was not explicitly represented.

4 Results

Accuracies for the various parsers are shown in Table 4, with the highest score for each construction in bold. Enju and C&C are the top performers, operating at roughly the same level of accuracy across most of the constructions. Use of the C&C question model made a huge difference for the *wh*-object construction (81.2% vs. 27.5%), showing that adaptation techniques specific to a particular

construction can be successful (Rimell and Clark, 2008).

In order to learn more from these results, in Section 5 we analyse the various errors made by the C&C parser on each construction. The conclusions that we arrive at for the C&C parser we would also expect to apply to Enju, on the whole, since the design of the two parsers is so similar. In fact, some of the recommendations for improvement on this corpus, such as the need for a better parsing model to make better attachment decisions, are parser independent.

The poor performance of RASP on this corpus is clearly related to a lack of subcategorisation information, since this is crucial for recovering extracted arguments. For Stanford, incorporating the trace information from the PTB into the statistical model in some way is likely to help. The C&C and Enju parsers do this through their respective grammar formalisms. Our informal impression of the DCU post-processor is that it has much of the machinery available to recover the dependencies that the Enju and C&C parsers do, but for some reason which is unclear to us it performs much worse.

5 Analysis of the C&C Parser

We categorised the errors made by the C&C parser on the development data for each construction. We chose the C&C parser for the analysis because it was one of the top performers and we have more knowledge of its workings than those of Enju.

The C&C parser first uses a supertagger to assign a small number of CCG lexical categories (essentially subcategorisation frames) to each word in the sentence. These categories are then combined using a set of combinatory rules to build a CCG derivation. The parser uses a log-linear probability model to select the highest-scoring derivation (Clark and Curran, 2007). In general, errors in dependency recovery may occur if the correct lexical category is not assigned by the supertagger for one or more of the words in a sentence, or if an incorrect derivation is chosen by the parsing model.

For unbounded dependency recovery, one source of errors (labeled **type 1** in Table 5) is the wrong lexical category being assigned to the word (normally a verb or preposition) governing the extraction site. In *these testaments that I would submit here*, if *submit* is assigned a category for an intransitive rather than transitive verb, the verb-object relation will not be recovered.

	1a	1b	1c	1d	2	3	Errs	Tot
ObjRC			6		5	2	13	20
ObjRed	2		1	1	1	3	8	23
SbjRC					8	1	9	43
Free	1					1	2	22
ObjQ			2	2			4	25
RNR			2	1	7	3	13	28
SbjEmb	3	2	1			4	10	13
Subtotal	6	2	12	4				
Total	24				21	14	59	174

Table 5: Error analysis for C&C. *Errs* is the total number of errors for a construction, *Tot* is the number of dependencies of that type in the development data.

There are a number of reasons why the wrong category may be assigned. First, the lexicon may not contain enough information about possible categories for the word (**1a**), or the necessary category may not exist in the parser's grammar at all (**1b**). Even if the grammar contains the correct category and the lexicon makes it available, the parsing model may not choose it (**1c**). Finally, a POS-tagging error on the word may mislead the parser into assigning the wrong category (**1d**).[2]

A second source of errors (**type 2**) is attachment decisions that the parser makes independently of the unbounded dependency. In *Morgan ... carried in several buckets of water from the spring which he poured into the copper boiler*, the parser assigns the correct categories for the relative pronoun and verb, but chooses *spring* rather than *buckets* as the head of the relativized NP (i.e. the object of *pour*). Most attachment errors involve prepositional phrases (PPs) and coordination, which have long been known to be areas where parsers need improvement.

Finally, errors in unbounded dependency recovery may be due to complex errors in the surrounding parse context (**type 3**). We will not comment more on these cases since they do not tell us much about unbounded dependencies in particular.

Table 5 shows the distribution of error types across constructions for the C&C parser. Subject relative clauses, for example, did not have any errors of type 1, because a verb with an extracted

[2]We considered an error to be type 1 only when the category error occurred on the word governing the extraction site, except in the subject embedded sentences, where we also included the embedding verb, since the category of this verb is key to dependency recovery.

subject does not require a special lexical category. Most of the errors here are of type 2. For example, in *a series of pipes and a pressure-measuring chamber which record the rise and fall of the water surface*, the parser attaches the relative clause to *chamber* but not to *series*.

Subject embedded sentences show a different pattern. Many of the errors can be attributed to problems with the lexicon and grammar (1a and 1b). For example, in *shadows that they imagined were Apaches*, the word *imagined* never appears in the training data with the correct category, and so the required entry is missing from the lexicon.

Object extraction from a relative clause had a higher number of errors involving the parsing model (1c). In *the first carefree, dreamless sleep that she had known*, the transitive category is available for *known*, but not selected by the model.

The majority of the errors made by the parser are due to insufficient grammar coverage or weakness in the parsing model due to sparsity of head dependency data, the same fundamental problems that have dogged automatic parsing since its inception. Hence one view of statistical parsing is that it has allowed us to solve the easy problems, but we are still no closer to a general solution for the recovery of the "difficult" dependencies. One possibility is to create more training data targeting these constructions – effectively "active learning by construction" – in the way that Rimell and Clark (2008) were able to build a question parser. We leave this idea for future work.

6 Discussion

Unbounded dependencies are rare events, out in the Zipfian "long tail". They will always constitute a fraction of a percent of the overall total of head-dependencies in any corpus, a proportion too small to make a significant impact on global measures of parser accuracy, when expressive parsers are compared to those that merely approximate human grammar using finite-state or context-free covers. This will remain the case even when such measures are based on dependencies, rather than on parse trees.

Nevertheless, unbounded dependencies remain highly significant in a much more important sense. They support the constructions that are central to those applications of parsing technology for which precision is as important as recall, such as open-domain question-answering. As low-power approximate parsing methods improve (as they must if they are ever to be usable at all for such tasks), we predict that the impact of the constructions we examine here will become evident. No matter how infrequent object questions like "What do frogs eat?" are, if they are answered as if they were subject questions ("Herons"), users will rightly reject any excuse in terms of the overall statistical distribution of related bags of words.

Whether such improvements in parsers come from the availability of more human-labeled data, or from a breakthrough in unsupervised machine learning, we predict an imminent "Uncanny Valley" in parsing applications, due to the inability of parsers to recover certain semantically important dependencies, of the kind familiar from humanoid robotics and photorealistic animation. In such applications, the closer the superficial resemblance to human behavior gets, the more disturbing subtle departures become, and the more they induce mistrust and revulsion in the user.

7 Conclusion

In this paper we have demonstrated that current parsing technology is poor at recovering some of the unbounded dependencies which are crucial for fully representing the underlying predicate-argument structure of a sentence. We have also argued that correct recovery of such dependencies will become more important as parsing technology improves, despite the relatively low frequency of occurrence of the corresponding grammatical constructions. We also see this more focused parser evaluation methodology — in this case construction-focused — as a way of improving parsing technology, as an alternative to the exclusive focus on incremental improvements in overall accuracy measures such as Parseval.

Acknowledgments

Laura Rimell and Stephen Clark were supported by EPSRC grant EP/E035698/1. Mark Steedman was supported by EU IST Cognitive Systems grant IP FP6-2004-IST-4-27657 (PACO-PLUS). We would like to thank Aoife Cahill for producing the DCU data.

References

E. Black, S. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *HLT '91: Proceedings of the Workshop on Speech and Natural Language*, pages 306–311.

Rens Bod. 2003. An efficient implementation of a new DOP model. In *Proceedings of the 10th Meeting of the EACL*, pages 19–26, Budapest, Hungary.

Ted Briscoe, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of the Interactive Demo Session of COLING/ACL-06*, Sydney, Australia.

A. Cahill, M. Burke, R. O'Donovan, J. van Genabith, and A. Way. 2004. Long-distance dependency resolution in automatically acquired wide-coverage PCFG-based LFG approximations. In *Proceedings of the 42nd Meeting of the ACL*, pages 320–327, Barcelona, Spain.

Xavier Carreras, Michael Collins, and Terry Koo. 2008. Dynamic programming and the perceptron for efficient, feature-rich parsing. In *Proceedings of the Twelfth Conference on Natural Language Learning (CoNLL-08)*, pages 9–16, Manchester, UK.

John Carroll, Ted Briscoe, and Antonio Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. In *Proceedings of the 1st LREC Conference*, pages 447–454, Granada, Spain.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Meeting of the ACL*, pages 173–180, Michigan, Ann Arbor.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Meeting of the NAACL*, pages 132–139, Seattle, WA.

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Stephen Clark, Mark Steedman, and James R. Curran. 2004. Object-extraction and question-parsing using CCG. In *Proceedings of the EMNLP Conference*, pages 111–118, Barcelona, Spain.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Meeting of the ACL*, pages 16–23, Madrid, Spain.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th LREC Conference*, Genoa, Italy.

Daniel Gildea. 2001. Corpus variation and parser performance. In *Proceedings of the 2001 EMNLP Conference*, Pittsburgh, PA.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Liang Huang. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of the 46th Meeting of the ACL*, pages 586–594, Columbus, Ohio.

Mark Johnson. 2002. A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *Proceedings of the 40th Meeting of the ACL*, pages 136–143, Philadelphia, PA.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of Association for Computational Linguistics*, pages 423–430, Sapporo, Japan.

Roger Levy and Christopher Manning. 2004. Deep dependencies from context-free statistical parsers: correcting the surface dependency approximation. In *Proceedings of the 42nd Meeting of the ACL*, pages 328–335, Barcelona, Spain.

Dekang Lin. 1995. A dependency-based method for evaluating broad-coverage parsers. In *Proceedings of IJCAI-95*, pages 1420–1425, Montreal, Canada.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Conference*, pages 152–159, Brooklyn, NY.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Meeting of the ACL*, pages 91–98, Michigan, Ann Arbor.

Yusuke Miyao and Jun'ichi Tsujii. 2005. Probabilistic disambiguation models for wide-coverage HPSG parsing. In *Proceedings of the 43rd Meeting of the ACL*, pages 83–90, Michigan, Ann Arbor.

J. Nivre and M. Scholz. 2004. Deterministic dependency parsing of English text. In *Proceedings of COLING-04*, pages 64–70, Geneva, Switzerland.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of the HLT/NAACL Conference*, Rochester, NY.

Laura Rimell and Stephen Clark. 2008. Adapting a lexicalized-grammar parser to contrasting domains. In *Proceedings of the 2008 EMNLP Conference*, pages 475–484, Honolulu, Hawai'i.

Appendix E

A Bottom-Up Parsing Model of Local Coherence Effects

Emily Morgan (emily@ling.ucsd.edu)
Department of Linguistics, 9500 Gilman Drive #108
La Jolla, CA 92093, USA

Frank Keller (keller@inf.ed.ac.uk)
Mark Steedman (steedman@inf.ed.ac.uk)
School of Informatics, 10 Crichton Street
Edinburgh EH8 9AB, UK

Abstract

Human sentence processing occurs incrementally. Most models of human processing rely on parsers that always build connected tree structures. But according to the theory of Good Enough parsing (Ferreira & Patson, 2007), humans parse sentences using small chunks of local information, not always forming a globally coherent parse. This difference is apparent in the study of local coherence effects (Tabor, Galantucci, & Richardson, 2004), wherein a locally plausible interpretation interferes with the correct global interpretation of a sentence. We present a model that accounts for these effects using a wide-coverage parser that captures the idea of Good Enough parsing. Using Combinatory Categorial Grammar, our parser works bottom-up, enforcing the use of local information only. We model the difficulty of processing a sentence in terms of the probability of a locally coherent reading relative to the probability of the globally coherent reading of the sentence. Our model successfully predicts psycholinguistic results.

Keywords: sentence processing; parsing complexity; local coherence; Good Enough parsing; Combinatory Categorial Grammar

Introduction

A major topic of inquiry in cognitive science is the process by which people produce and comprehend sentences. Human sentence processing is known to proceed incrementally: people construct syntactic and semantic interpretations gradually as a sentence unfolds, rather than waiting until after the whole sentence has been received. But although we know that syntactic information becomes available progressively while comprehending a sentence, it is still an open question to what extent decisions made early in the parsing process can constrain later decisions.

One phenomenon that can shed light on this question is local coherence effects. Local coherence effects arise when a sentence includes a substring with a plausible local interpretation that is incompatible with the global interpretation. (In other words, the interpretation is merely locally coherent, but not globally coherent.) A typical example (from Tabor, Galantucci, & Richardson, 2004) is:

(1) **A/R**: The coach smiled at the player tossed a frisbee.

A typical reader, seeing this sentence for the first time, will find it difficult to understand and will likely judge it to be ungrammatical. But this difficulty is unexpected in light of similar sentences:

(2) **U/R**: The coach smiled at the player thrown a frisbee.

(3) **A/U**: The coach smiled at the player who was tossed a frisbee.

(4) **U/U**: The coach smiled at the player who was thrown a frisbee.

These four sentences, all intended to be close paraphrases of one another, illustrate a puzzle: while the majority of readers reject (1), they accept (3) and (4), with mixed results for (2). These sentence differ on two dimensions: the past participle can be Ambiguous (such as *tossed*, which can be a past participle or a past tense form) or Unambiguous (such as *thrown*), and the relative clause can be Reduced (without *who was*) or Unreduced (with *who was*). Neither of these alternations generally changes the grammaticality of a sentence, so we would naively predict that if (4) is acceptable, then (1) is as well. Our challenge is to explain why this naive prediction is wrong. Intuitively, it seems that the local coherence of the substring *the player tossed a frisbee* in (1) as a plausible complete sentence is distracting from its globally correct interpretation as an object with a relative clause.

Tabor, Galantucci, and Richardson demonstrate the existence of local coherence effects as a psycholinguistic phenomenon in two different studies: in the first, they find increased reading times at the ambiguous past participle in (1). They present subjects with sentences from 20 sets of sentences like those seen above and measure reading times for each word using self-paced reading. In this methodology, longer reading times are taken to indicated increased processing difficulty. As expected based on previous studies (e.g. Ferreira & Clifton, 1986), they find substantially increased reading times for the Reduced cases as compared to the Unreduced cases, both on the past participle (e.g. *tossed*) and on the following word. Moreover, they find an unexpected interaction between Ambiguity and Reducedness: while the A/U reading times are not significantly different from the U/U reading times, the A/R reading times are substantially increased relative to the U/R reading times. This superadditive difficulty of the A/R condition is the signature of a local coherence effect.

In the second experiment, Tabor, Galantucci, and Richardson replicate the first using a grammaticality judgement task.

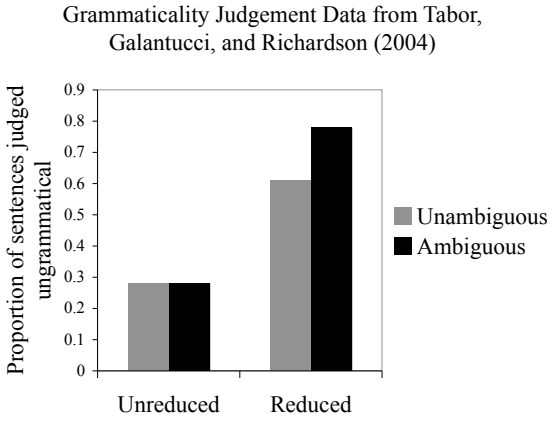


Figure 1: Grammaticality judgement data from Tabor, Galantucci, and Richardson (2004). The signature of a local coherence effect is the superadditive proportion of ungrammatical judgements in the Ambiguous/Reduced condition.

They find decreased acceptance of Reduced sentences as grammatical, with an interaction between Ambiguity and Reducedness such that A/R sentences are judged unacceptable superadditively often (see Figure 1). Once again, decreased acceptability judgements are taken to indicate processing difficulty.

Note that sentences in the A/R condition are not just standard garden path sentences. In a standard garden path sentence, the disambiguating information comes after the reader has already been led astray. In contrast, in sentences such as (1), the disambiguating information comes at the beginning of the sentence. Thus the reader in theory already knows that *tossed* cannot be a past tense form and must be a past participle. Yet despite that, these sentences cause processing difficulty.

A model of human sentence processing should be able to predict the difficulty of sentences with local coherence effects. However, most existing models cannot. In particular, most standard theories of parsing assume that that all accrued knowledge from the parsing process is taken into account at all times. Models following this assumption can straightforwardly account for standard garden paths because there is nothing inconsistent about initially misinterpreting a sentence before having access to the disambiguating information. But these models cannot take the same position in accounting for local coherence effects: when the disambiguating information has already been seen and *smiled* has already been recognized as the main verb of the sentence, they cannot entertain the inconsistent possibility that *tossed* is also a main verb. Computational implementations of wide-coverage parsers generally also make this assumption of global consistency (e.g. Roark, 2001; Sturt, Costa, Lombardo, & Frasconi, 2003; Demberg & Keller, 2008). For many applications, this assumption may be convenient. But for a parser to be credible as a model of human sentence processing, it must be able to predict these psycholinguistic effects, which requires relaxing this assumption.

An alternate theory of sentence processing is Ferreira and colleagues' *Good Enough* (GE) parsing. Ferreira and Patson (2007) describe GE parsing:

> People compute local interpretations that are sometimes inconsistent with the overall sentence structure, indicating that the comprehension system tries to construct interpretations over small numbers of adjacent words whenever possible and can be lazy about computing a more global structure and meaning.

The GE theory of parsing asserts that people do not construct full representations for sentences the majority of the time. Rather, they construct just enough to complete the task at hand, only constructing a further representation if necessary. Moreover, because people base their first-pass constructions on local information and generally construct only partial parse trees, these partial parses may contradict one another. A GE parsing account can thus easily account for local coherence effects. We will develop a computational model of why local coherence effects arise within the framework of GE parsing.

Previous Models of Local Coherence Effects

Two models have previously attempted to account for local coherence effects: Levy (2008) uses a noisy-channel model to argue that because there is uncertainty in linguistic input, the parse of a sentence should be modeled as a probability distribution over a set of candidate sentences (including the intended sentence and its near-neighbors). Given such a probability distribution, the effect of reading each word can be modeled and quantified in terms of a belief update. Levy predicts that a larger change in beliefs will correspond to greater processing difficulty and longer reading times. This in turn predicts local coherence effects because the rarer sentences provoke larger changes in belief.

Levy's model only considers fully connected and grammatical (partial) parses as candidates, thus it does not capture the intuition of GE parsing. An additional limitation of the model is that due to the computational load of calculating near-neighbors, it has only been implemented using a toy Probabilistic Context Free Grammar (PCFG), rather than a richer, wide-coverage language model.

The other previously existing model of local coherence effects comes from Bicknell and Levy (2009). They again model local coherence effects as arising from belief updates. Specifically, they model them as the consequences of an update from a bottom-up prior belief to a posterior belief that takes top-down information into account. They thus predict processing difficulty in the case of locally coherent substrings because the bottom-up statistics make strong predictions about the category of the substrings, which are then contradicted by top-down information.

This model begins to capture the idea of GE parsing by looking at substrings of different lengths. However, it has no way to integrate the information it receives from these different substring lengths because evaluating these substrings is post hoc, not an actual part of the parsing process. Additionally, like Levy's (2008) model, it has only been implemented using a toy PCFG.

Thus there is currently no general, wide-coverage model of human parsing that implements a GE parsing strategy. Computational models of local coherence effects have instead had to account for the phenomenon indirectly, either through a noisy channel model or by predicting the effects without actually simulating the parsing process, and have been confined to parsing with small toy grammars. We will develop a model to address these shortcomings.

A New Model of Local Coherence Effects

Our goal is to model the process by which local coherence effects emerge as the result of Good Enough parsing, within the context of a wide-coverage parser. In the example sentence *The coach smiled at the player tossed a frisbee*, our intuition is that processing difficulty arises from the locally coherent reading of *the player tossed a frisbee*, which distracts from the globally coherent reading. Our model will capture this intuition by using a strictly bottom-up parser to remove the top-down influence of non-local constraints.

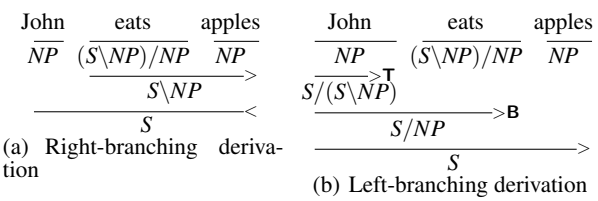
Strictly bottom-up parsing is frequently rejected as a plausible model for human parsing because, it is claimed, it does not allow for incremental interpretation. The standard argument says that a clause can only be interpreted when it is seen in full (i.e., at the end of a constituent). But in a strictly right-branching language, this means that nothing can be interpreted until the very end of the sentence because only then is any constituent completed.

To overcome this objection, our parser uses the Combinatory Categorial Grammar (CCG) formalism to represent linguistic structures. CCG was specifically designed to allow for incremental bottom-up parsing by using a more flexible notion of constituents.

Combinatory Categorial Grammar

Combinatory Categorial Grammar is a grammar formalism based on Categorial Grammar (CG). We base our description of it here on Steedman (2000).

CCG revolves around functional categories and rules for combining them. Categories can be either functions or arguments and are defined recursively: Base categories such as *S* and *NP* represent arguments. Functions are of the form α/β or $\alpha\backslash\beta$, where α and β are categories. To the right of the slash is the argument of the function, and to the left is its result. The direction of the slash indicates the directionality of composition: / means the argument is to the right and \ means the argument is to the left. An English verb phrase, for example, will have the category $S\backslash NP$, indicating that it combines with an NP on its left and results in a sentence. We also allow a finite set of features on our base categories, such as person, number, and gender on *NP*s. These are notated as e.g. *NP*[3sf].



(a) Right-branching derivation

(b) Left-branching derivation

Figure 2: Right- and left-branching CCG derivations for the sentence *John eats apples*. $(S\backslash NP)/NP$ is the CCG category for a transitive verb. Without type-raising, *eats* can only combine with *apples*, yielding the typical right-branching derivation in (a). With type-raising, *John* can combine immediately with *eats*, yielding the left-branching derivation in (b).

A CCG derivation uses rules to combine categories. Pure CG relies on two rules, named $>$ and $<$, to combine categories:

(5) $\quad X/Y \quad Y \rightarrow X \quad (>)$

(6) $\quad Y \quad X\backslash Y \rightarrow X \quad (<)$

CCG introduces further combinatory rules that allow for more flexible notions of constituency than other grammar formalisms. In particular, it includes two lexical *type-raising* rules, named $>\mathbf{T}$ and $<\mathbf{T}$:

(7) $\quad X \rightarrow T/(T\backslash X) \quad (>\mathbf{T})$

(8) $\quad X \rightarrow T\backslash(T/X) \quad (<\mathbf{T})$

In these rules—which are here shown in the derivation, but in fact operate in the lexicon—T can be any lexical category taking X as argument. For instance, we could use $>\mathbf{T}$ to type-raise *NP* to $S/(S\backslash NP)$. Applying this rule limits the other categories the NP can combine with. Intuitively, we can think of the output of this rule as similar to an *NP* with nominative case-marking. It specifies not just that the word or phrase in question is a noun, but that it is a subject which must combine with a predicate.

These type raising rules allow us to parse a sentence incrementally by forming nontraditional constituents, leading to left-branching derivations (see Figure 2). CCG thus allows each new word of the input to be incorporated into the existing constituent structure as it is encountered, which makes incremental bottom-up parsing possible.

The Model

We take a bottom-up CCG parser as the basis of our model of human sentence processing. In order to predict processing difficulty caused by local coherence effects, we need a linking hypothesis to specify the relation between the parser output and psycholinguistic measures such as grammaticality judgements or reading times. Our linking hypothesis should embody the theory of Good Enough parsing, focusing on in-

terpretations of local substrings.

We adapt a model proposed by Jurafsky (1996) to predict garden path effects. To make graded predictions, rather than categorical distinctions, we will adopt a probabilistic framework, and consider the probabilities of various substrings of a sentence. In particular, we could consider either the inside probability $P(S \rightarrow \text{substring})$ (alternately written as $P(\text{substring} \mid S)$) or the inverse probability $P(S \mid \text{substring})$. We do not know of a computationally tractable way to calculate $P(S \mid \text{substring})$ from our parser. Calculating the inside probability, on the other hand, is a fundamental part of the parsing process. It is most parsimonious to base our model on the inside probabilities that are already being calculated.

Our intuition is that if an incorrect interpretation of a substring is highly plausible relative to the correct interpretation of the sentence, then it will cause processing difficulty. In a sentence such as *The coach smiled at the player tossed a frisbee*, the substring that we expect to cause difficulty is the locally coherent substring *the player tossed a frisbee*. We thus consider the ratio:

$$\frac{P(S \rightarrow \textit{the player tossed a frisbee})}{P(S \rightarrow \textit{The coach smiled at the player tossed a frisbee})}$$

In this case, the ratio will be high because *The player tossed a frisbee* is a relatively likely sentence. In the other three cases, the ratio will be low because none of the following are very plausible sentences:

(9) the player thrown a frisbee

(10) the player who was tossed a frisbee

(11) the player who was thrown a frisbee

Although in theory this ratio could be as low as 0, in practice this does not occur because there is generally some (low probability) way to parse each phrase as a sentence. We take this ratio as a measure of processing difficulty.

Implementation

We implement our model using a Combinatory Categorial Grammar parser based on the Cocke-Kasami-Younger (CKY) algorithm. This algorithm was originally developed for Context Free Grammars and uses dynamic programing to parse from the bottom up: given a sentence, it first calculates the probabilities of all ways to generate each word using a rule $X \rightarrow word$. For each potential pair of categories X_1 and X_2 that could have generated adjacent words w_1 and w_2, it then calculates the probabilities of all ways to generate that pair using a rule $X_3 \rightarrow X_1X_2$. This allows us to calculate the inside probability $P(X_3 \rightarrow w_1w_2)$. Continuing iteratively, we can calculate the inside probabilities of all substrings of the sentence.

We used a modified version of the StatOpenCCG parser, developed by Christodoulopoulos (2008), which is itself an extension of the OpenCCG parser (White, 2008). StatOpenCCG implements a statistical version of the CKY algorithm which operates using a generative head-dependency model over CCG categories: From the parent (starting with a ROOT node), a head is generated with a certain probability. Then its sisters are generated with probability conditioned on the head category, the sister's direction from the head, and whether it is adjacent to the head. Although the number of CCG categories is theoretically infinite, our parser is constrained to only use categories that have appeared in the training data set. With this constraint, the runtime of the parser is bounded by $O(n^3)$. The parser has been trained on sections 1 through 22 of the CCGbank (Hockenmaier, 2003), a CCG version of the Penn treebank.

Our experiments use two different lexicons. The first lexicon is that taken from sections 1 through 22 of the CCGbank. However, this lexicon is too small to parse the majority of the sentences we wish to consider. To obtain a larger lexicon, we parsed six months of the New York Times (comprising approximately 50 million word tokens) taken from the Gigaword corpus (Graff, 2003). Sentences from the corpus were passed through the RASP tokenizer (Briscoe, Carroll, & Watson, 2006) and then parsed using the C&C CCG parser (Curran, Clark, & Bos, 2007). This state-of-the-art parser obtains labelled precision of 84.8% and labelled recall of 84.5% on section 23 of the CCGbank. It is extremely fast and provides the best parse accuracy from a CCG parser, making it convenient for obtaining large amounts of data to construct a larger lexicon. (However, it is not a cognitively plausible parser, as it relies on its supertagger and other cognitively implausible tricks to speed its parsing.) From this parsed sample, we extracted the lexicon for use in the StatOpenCCG parser (with the statistical parsing model over categories trained as before on CCGbank data). Although this lexicon of course contains quite a few errors, we verify that it nonetheless parses our test sentences correctly, placing the correct parses among the top results.

Experiments

We present two sets of experiments in which we test our model against the results from Tabor, Galantucci, and Richardson (2004). The first uses a small but high-quality lexicon to parse two test cases. The second uses a larger, error-ridden lexicon to parse a larger set of sentences. Recall that Tabor, Galantucci, and Richardson's (2004) study used 20 sets of sentences like those in (1)–(4).

Experiment 1: Test Cases using the CCGbank Lexicon

Because CCGbank is derived from a human-annotated treebank, the quality of the lexicon it yields is high. Nevertheless, it is small in comparison to human lexicons, and the passive relative constructions we are investigating are sparsely represented. In fact, the CCGbank lexicon contains only two words which are unambiguous ditransitive passive participles (i.e., $(S[\text{pss}]\backslash NP)/NP$ but not $(S[\text{dcl}]\backslash NP)/NP$—where [pss] indicates a past participle used in a passive construction, and [dcl] indicates a declarative sentence). These two words are

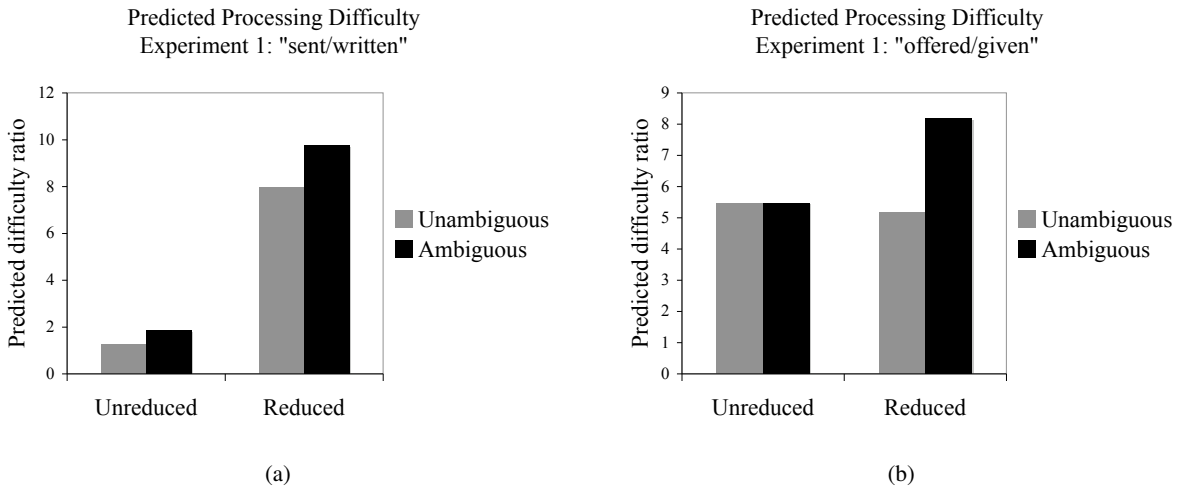


Figure 3: Results from Experiment 1, two test cases using the high-quality CCGbank lexicon. In both sets of sentences, the A/R case displays the correct pattern of superadditive difficulty.

written and *given*. Using these words, we construct two sentence sets, based on sentences used by Tabor, Galantucci, and Richardson:

(12) He questioned a congressman (who was) sent/written a letter.

(13) He addressed the woman (who was) offered/given a beer.

All words in these sentences are in the CCGbank lexicon. We parse them using our high-quality lexicon.

Results For these sentences, we obtain the predicted ratios:

$$\frac{\text{P}(S \rightarrow \text{locally coherent substring})}{\text{P}(S \rightarrow \text{whole sentence})}$$

Results are in Table 1 and Figure 3. We compare our results to the grammaticality judgements from Tabor, Galantucci, and Richardson (see Figure 1).

As we see in Figure 3(a), the set of sentences (12) displays the correct pattern of superadditive difficulty in the A/R case. While there is little difference in difficulty between the A/U and U/U conditions, there is a marked increase to the U/R condition, and a superadditive increase to the A/R condition. This mirrors the pattern seen in Tabor, Galantucci, and Richardson's grammaticality judgements.

We see the same superadditive pattern of difficulty in our results for the set of sentences (13), shown in Figure 3(b). Somewhat surprisingly, the U/R condition is in fact predicted to be marginally easier than the Unreduced sentences in this set. This may be because *given* is an extremely common word. Although it is unambiguous in that it cannot be a past tense, it is in fact a highly ambiguous word, with 18 entries in the CCGbank lexicon. For instance, it can serve as a preposition, as in *Given the weather, I will stay inside today*. Regardless of this slight puzzle, the A/R case displays the correct pattern of superadditive difficulty.

Table 1: Predicted difficulty ratios from all experiments, alongside grammaticality judgements from Tabor, Galantucci, and Richardson (2004).

Type	TG&R	Exp1: written	Exp1: given	Exp2
U/U	.28	1.27	5.45	5.74
A/U	.28	1.85	5.46	8.46
U/R	.61	7.96	5.16	11.60
A/R	.78	9.76	8.18	12.34

Experiment 2: Using the Gigaword Lexicon

Using the Gigaword lexicon, we are able to parse 13 out of the 20 sentences in the Tabor study. (Sentences were excluded only if their past participles were not present in the lexicon. All other vocabulary items are present.) We standardize all sentences to begin with a pronoun. Additionally, for the sake of parsing efficiency, we do not include the *by* phrases that give the agent of the sentence. We further shorten two sentence sets in ways that do not affect the target part of the sentence.

Results Results from Experiment 2 are shown in Table 1 and Figure 4. We compare our results to the grammaticality judgements from Tabor, Galantucci, and Richardson (see Figure 1). We find the correct trend of difficulties, with the A/R condition most difficult, followed by U/R, followed by the two Unreduced cases. We do not find the exact pattern of superadditive difficulty in the A/R case, due to the fact that the A/U case is in fact predicted to be much more difficult than the U/U case, in contrast to the grammaticality ratings. Because the Gigaword lexicon is very error-prone, it is difficult

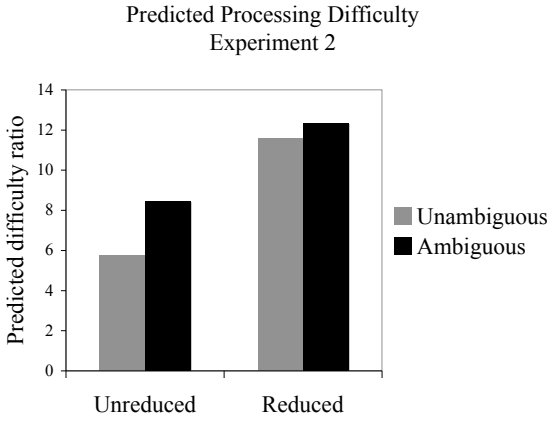


Figure 4: Experiment 2 results. We find the expected pattern of difficulty, but, due to the inflated predicted difficulty of the U/R case, do not see superadditive difficulty in the A/R case.

to draw any firm conclusions from this quirk in our results. However, we note that the A/R case is correctly predicted to be substantially more difficult than either of the Unreduced cases.

Conclusion

We have presented a model of local coherence effects using a wide-coverage bottom-up Combinatory Categorial Grammar parser. Our model can accurately predict which sentences humans will have difficulty in processing; specifically, it predicts the local coherence effects found by Tabor, Galantucci, and Richardson (2004). Our results support the psycholinguistic plausibility of CCG and the Good Enough theory of parsing by demonstrating that a parser that uses bottom-up local information can both perform well as a wide-coverage parser and predict specific psycholinguistic results.

Interestingly, the architecture of our version of the GE parser differs from Ferreira's original proposal. Ferreira (2003) proposes that GE parsing occurs via two separate strategies: one "algorithmic" and one "heuristic". In contrast, our parser does not include this separation: all analyses, both local and global, are produced by a uniform algorithm, and all are heuristically evaluated using the parsing model. This integration of strategies is a strength of our model, as it demonstrates how local coherence effects could emerge naturally as an inherent part of the parsing process.

In future work, we would like to make not just sentence-level predictions but word-by-word reading time predictions. Given that we have an entire parse chart, such predictions should be possible. We are currently choosing inside probabilities from two cells in the parse chart to compare, based on outside knowledge of where processing difficulty is likely to arise. We could do something similar for every cell in the chart, considering the inside probability of the substring it spans relative to the probability of the sentence as a whole. With word by word predictions, we could model reading time data as well as grammaticality judgement data. Such a model would be applicable to a wide range of psycholinguistic data beyond local coherence effects.

Acknowledgments

This work was supported by EU IST Cognitive Systems IP FP6-2004-IST-4-27657 "Paco-Plus".

References

Bicknell, K., & Levy, R. (2009). A model of local coherence effects in human sentence processing as consequences of updates from bottom-up prior to posterior beliefs. In *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT) 2009 Conference* (pp. 665–673). Boulder, CO: Association for Computational Linguistics.

Briscoe, E., Carroll, J., & Watson, R. (2006). The second release of the RASP system. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions.* Sydney, Australia: Association for Computational Linguistics.

Christodoulopoulos, C. (2008). *Creating a natural logic inference system with Combinatory Categorial Grammar*. Master's thesis, University of Edinburgh.

Curran, J. R., Clark, S., & Bos, J. (2007). Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of the ACL 2007 Demonstrations Session (ACL-07 demo)* (pp. 29–32). Morristown, NJ: Association for Computational Linguistics.

Demberg, V., & Keller, F. (2008). A psycholinguistically motivated version of TAG. In *Proceedings of the 9th International Workshop on Tree Adjoining Grammars and Related Formalisms* (pp. 25–32). Tübingen.

Ferreira, F. (2003). The misinterpretation of noncanonical sentences. *Cognitive Psychology*, *47*, 164–203.

Ferreira, F., & Clifton, C., Jr. (1986). The independence of syntactic processing. *Journal of Memory and Language*, *25*, 348–368.

Ferreira, F., & Patson, N. D. (2007). The 'Good Enough' approach to language comprehension. *Language and Linguistics Compass*, *1*(1–2), 71–83.

Graff, D. (2003). *English Gigaword.* Linguistic Data Consortium, Philadelphia. (DVD)

Hockenmaier, J. (2003). *Data and models for statistical parsing with Combinatory Categorial Grammar*. Doctoral dissertation, University of Edinburgh.

Jurafsky, D. (1996). A probabilistic model of lexical and syntactic access and disambiguation. *Cognitive Science: A Multidisciplinary Journal*, *20*(2), 137–194.

Levy, R. (2008). A noisy-channel model of rational human sentence comprehension under uncertain input. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).* Morristown, NJ: Association for Computational Linguistics.

Roark, B. (2001). Probabilistic top-down parsing and language modeling. *Computational Linguistics*, *29*(2), 249–276.

Steedman, M. (2000). *The syntactic process*. Cambridge, MA: The MIT Press.

Sturt, P., Costa, F., Lombardo, V., & Frasconi, P. (2003). Learning first-pass structural attachment preferences with dynamic grammars and recursive neural networks. *Cognition*, *88*, 133–169.

Tabor, W., Galantucci, B., & Richardson, D. (2004). Effects of merely local syntactic coherence on sentence processing. *Journal of Memory and Language*, *50*, 355–370.

White, M. (2008). *Open CCG: The OpenNLP CCG library.* (http://openccg.sourceforge.net/ [Online; accessed 27-July-2009])